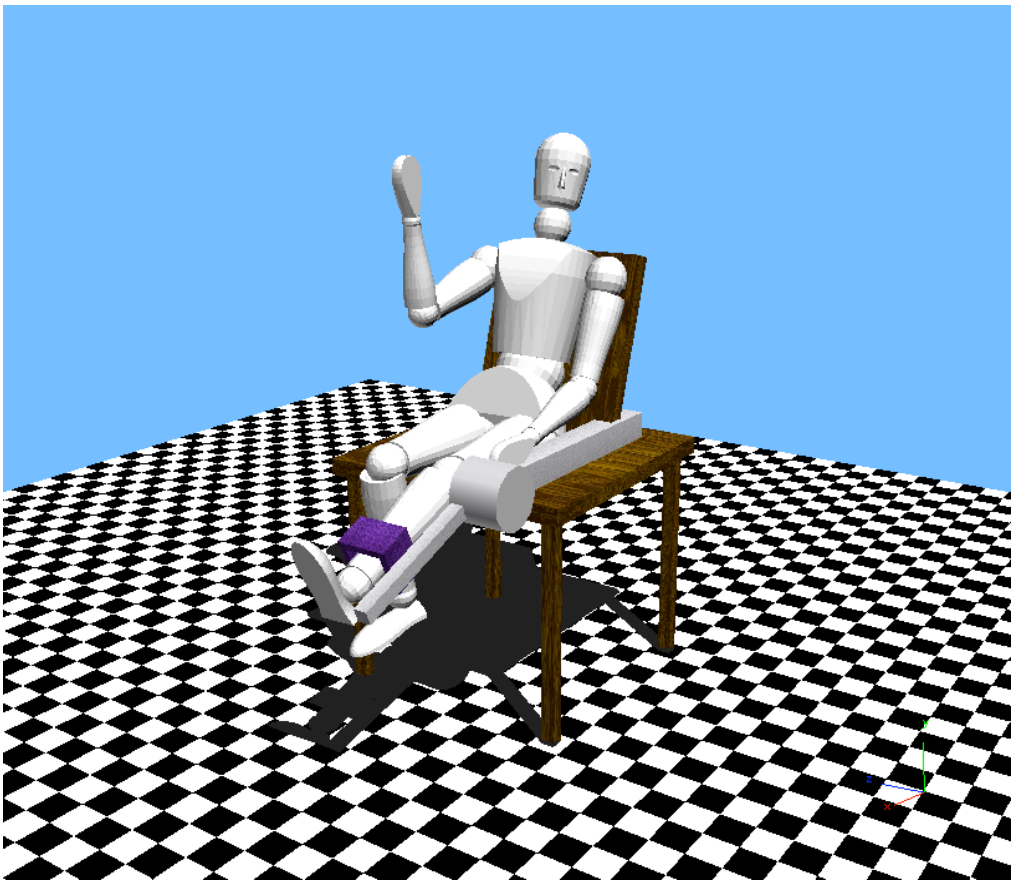
Projet de Semestre

Environnement virtuel de modélisation de mouvements humains et de stimulation musculaire



Candidat : Cédric Favre

Professeur : Reymond Clavel
Assistants : Yves Stauffer et Ludovic Righetti

25 juin 2007

Table des matières

1 Introduction **6**
- 1.1 Introduction . . . 6
- 1.2 Planning original . . . 9

2 Introduction à Webots **10**
- 2.1 La notion de ”monde” . . . 10
- 2.2 La notion de contrôleur . . . 10
- 2.3 La notion de plugin physique . . . 11
- 2.4 Interface Webots . . . 11
- 2.5 Fonctionnement . . . 12
  - 2.5.1 Déroulement de la simulation . . . 12
  - 2.5.2 Contrôle des servos . . . 13

3 Aspect Visuel : Creation d'un humanoïde sous Webots **15**
- 3.1 Modélisation . . . 15
  - 3.1.1 Convention pour le sens des axes . . . 15
  - 3.1.2 Manière de Modéliser . . . 15
  - 3.1.3 Données Biologiques et Hypothèses . . . 16
- 3.2 Mouvements . . . 18
  - 3.2.1 Génération des trajectoires . . . 18
  - 3.2.2 Suivi des trajectoires . . . 18
- 3.3 Interface graphique . . . 19
  - 3.3.1 Résultat Visuel . . . 19
  - 3.3.2 Adaptation dynamiques des amplitudes . . . 19
- 3.4 Programme de génération de modèles . . . 20

4 Mesure des couples lors de la marche **24**
- 4.1 But . . . 24
- 4.2 Problèmes . . . 24
- 4.3 Conditions de mesure . . . 25
- 4.4 Résultats . . . 25

5 Commande des articualtions en couple **29**
- 5.1 PID . . . 29
  - 5.1.1 Définition . . . 29
  - 5.1.2 Implémentation . . . 29
- 5.2 Integration du modèle de muscle . . . 30
  - 5.2.1 Modèle . . . 30

6 Simulation avec l'orthèse du genou **32**
6.1 Création de l'orthèse . . . 32
6.2 Mesure du couple produit par le genou sur l'orthèse . . . 32
6.3 Problèmes rencontrés . . . 32
6.4 Résultats . . . 34
6.4.1 Cas simple statique . . . 34
6.4.2 Couple généré par la masse en fonction de la position . . . 36
6.4.3 Couple généré par le muscle stimulé avec une intensité de 35 mA en fonction de la position . . . 37
6.4.4 Données sur patient réels . . . 37
6.4.5 Comparaison entre couple mesuré et couple théorique généré . . . 39
6.4.6 Analyse . . . 39
6.4.7 Remarque . . . 39

7 Améliorations et Travail futur **43**

8 Conclusion **44**

Bibliographie **45**

9 Annexes **46**
9.1 Contenu du CD-ROM . . . 46

10 Outils de travail **47**

11 Appendice **48**

LABORATOIRE DE SYSTEMES ROBOTIQUES (LSRO)

Adresse postale : **EPFL LSRO**
ME3, Station 9
CH 1015 Lausanne

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

PROJET DE SEMESTRE ETE 2007

Titre: Environnement virtuel de modélisation de mouvements humains et de stimulation musculaire

Candidat: Cédric FAVRE **Section:** Informatique

Professeur: Reymond Clavel

Assistants: Ludovic Righetti, Yves Stauffer

Introduction.

Dans le cadre de la rééducation de blessés médullaires nous sommes amenés à effectuer des mesures biomécaniques. Notamment des membres inférieurs et du bassin. La représentation ainsi que l'analyse de ces résultats peuvent s'avérer complexes. Pour faciliter ces opérations, il s'agira dans ce projet d'employer un logiciel de réalité virtuelle (Webots) afin de réaliser un homme virtuel, incluant des longueurs de segments paramétrables ainsi que des masses physiologiques ajustables. Cet humain virtuel sera utilisé pour la visualisation et pour la validation des mesures. Dans un deuxième temps il s'agira d'inclure le modèle du muscle afin de pouvoir effectuer des simulations de stimulation musculaire. Sur une orthèse du genou, puis sur le MotionMaker.

Travail demandé.

Visualisation (30%) :

- Créer un être humain virtuel sous Webots. Les longueurs de segments ainsi que leurs masses seront calquées sur un homme « standard ». Les aspects visuels seront importés depuis ProEngineer (modèle humain déjà développé).

- Contrôler en position les articulations de la hanche, du genou et de la cheville. Un générateur de marche développé sous Matlab contient toutes les informations nécessaires (coefficients de Fourier).

- Mettre en évidence les mouvements selon chaque degré de liberté (au moyen de « sliders » par exemple).

Modélisation (70%) :

- Commander les articulations en position et mesurer les couples nécessaires au mouvement, ceci afin de valider notre modèle développé dans Matlab.

- Intégrer un modèle du muscle au niveau de l'articulation. En clair il s'agira de commander les articulations en couple. Ce dernier dépendra de l'intensité de la stimulation électrique et de la position. Le modèle développé par P. Métrailler sera utilisé. Prévoir la possibilité d'inclure des paramètres liés à la vitesse de l'articulation également.

- Valider la simulation en créant une orthèse du genou virtuelle simplifiée qui permettra une comparaison avec les résultats réels. En cas de succès il sera possible de simuler le MotionMaker AVEC la stimulation musculaire.

Le code devra être modulaire et réutilisable : C++ avec commentaires.

Remarques.

Un plan de travail détaillé écrit sera remis aux responsables du projet avant le 26 mars 2007.

Un rapport dactylographié **(noir – blanc)** comprenant en son début l'énoncé du travail suivi d'un résumé d'une page (forme standardisée, noir et blanc; 2 exemplaires du résumé seront également joints en feuilles volantes et la version informatique Word sera adressée au professeur responsable) sera remis le 25 juin 2007 à 16 heures au plus tard en 3 exemplaires. Une photo du candidat (format passeport) sera intégrée à la page de garde du rapport. Ces documents seront également gravés sur un CD (forme PDF limitée à 3 MBytes pour le rapport sans les annexes, Word pour le résumé) et remis à l'assistant responsable. Ces documents seront inclus par le LSRO sur son site web pour autant qu'il n'y ait pas de problème de confidentialité et que le niveau de qualité soit suffisant.
Une défense de 45 minutes (environ 25 minutes de présentation et démonstration, plus 20 minutes de réponses aux questions) aura lieu dans la période du 26 juin au 4 juillet 2007.

Le professeur responsable: Assistants responsables:

R. Clavel L. Righetti Y.Stauffer

Lausanne, le 08.03.2007

Environnement virtuel de modélisation de mouvements humains et de stimulation musculaire.

Cédric Favre, Informatique

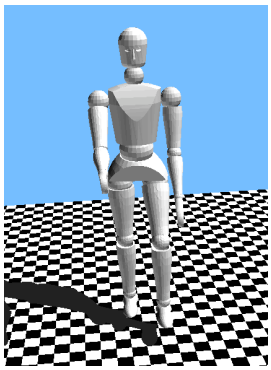
Assistants: *Yves Stauffer et Ludovic Righetti*

Professeur: *Reymond Clavel*

But :

Ce projet s'inscrit dans le cadre du travail mené par le LSRO autour de la Cyberthèse qui vise la rééducation de patients paraplégiques.

Le but principal de ce projet est de vérifier que Webots (un logiciel de réalité virtuel) est capable de simuler fidèlement un système aussi complexe que les membres inférieurs d'un humain en mouvement et pour se faire de créer un modèle humanoïde qui sera soumis à divers tests de validation.



Humanoïde en train de « marcher »

Déroulement

Le projet est composé de trois phases distinctes :

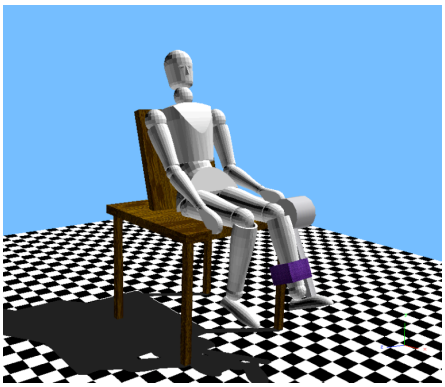
- Modélisation visuelle de l'humanoïde.
- Mesures des couples aux articulations, afin de vérifier l'exactitude de la simulation physique.
- Commande des articulations en couple et validation de la simulation conjointe de l'orthèse du genou et de l'humanoïde.

Résultats Principaux :

Le robot humanoïde a été modélisé avec succès. Il suit des trajectoires de marche réelles. Le modèle possède une interface graphique permettant de mettre en évidence divers mouvements lors de la marche.

Les couples mesurés aux articulations sont similaires en forme à ceux générés par le modèle Matlab de Yves Allemand, Des différences d'intensité sont cependant mesurées. Ces dernières semblent principalement dues au fait que les deux modèles utilisent des données biomécaniques différentes.

Le modèle de muscle présenté par le docteur P.Métraillé a été implémenté et testé par l'intermédiaire d'une simulation de l'orthèse du genou. Les résultats montrent le comportement souhaité, la seule différence avec le modèle original étant le moment passif induit par les ligaments et autres forces biomécaniques qui ne sont pas simulés dans le modèle Webots.



L'humanoïde et l'orthèse du genou.

Chapitre 1

Introduction

1.1 Introduction

L'assistance robotisée offre des possibilités de traitements thérapeutiques de mobilisation aux paralysés. Le LSRO travaille sur un projet nommé Cyberthèse [LSR] qui vise la rééducation de blessés médullaires associant orthèse et électrostimulation musculaire.

> La paralysie est un effet consécutif à une lésion du système nerveux central. Il en résulte une immobilité qui s'accompagne de complications médicales parfois importantes. On peut les prévenir en mobilisant les membres paralysés et en entretenant la musculature par des exercices de physiothérapie ou de fitness. A quoi s'ajoutent désormais des traitements basés sur l'électrostimulation neuromusculaire - ou Stimulation électrique fonctionnelle (SEF). Pour être efficace, un programme de rééducation doit stimuler les muscles selon des séquences et des efforts qui reproduisent le plus fidèlement possible les contractions musculaires naturelles. La cinématique et la dynamique des déplacements articulaires provoquées par électrostimulation doivent être les plus similaires possibles aux mouvements physiologiques. [LSR]

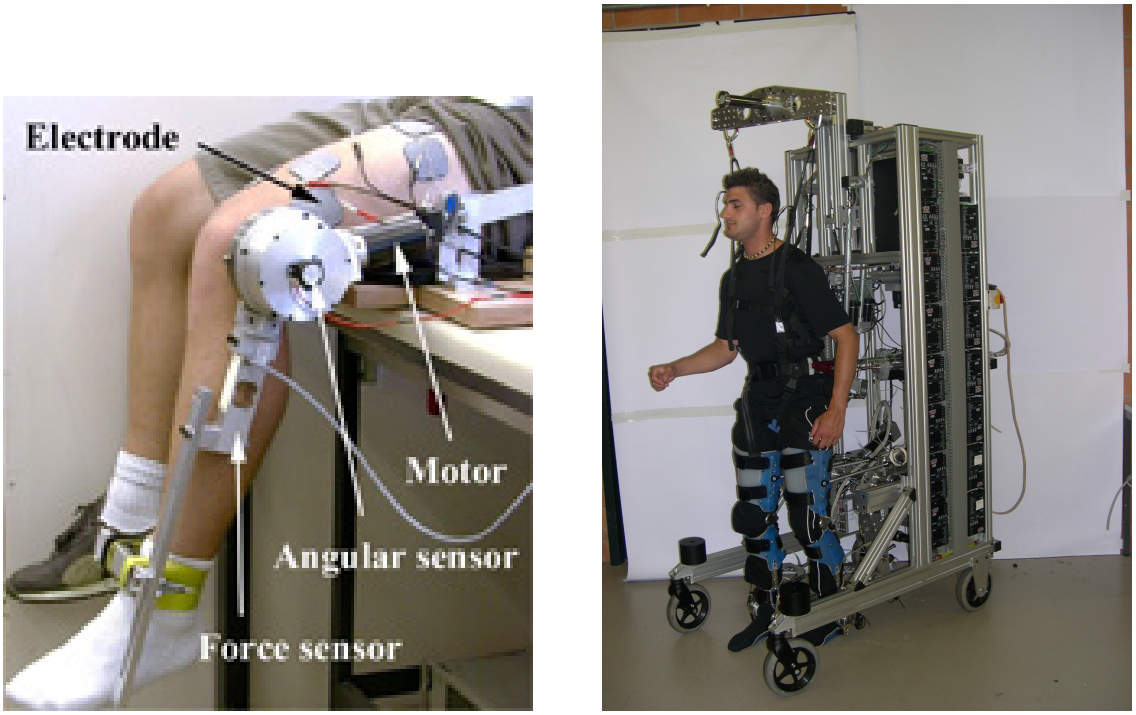
Le LSRO a créé plusieurs orthèses dont deux modèles sont présentés sur la figures 1.1. Le modèle 1.1(a) est une orthèse de genou, alors que le modèle 1.1(b) est le modèle le plus avancé actuellement.

Dans le cadre de ce projet, le LSRO est amené à effectuer des mesures biomécaniques. Notamment des membres inférieurs et du bassin. La représentation ainsi que l'analyse de ces résultats peuvent s'avérer complexes. Le but de ce projet est d'employer Webots afin de modéliser un humanoïde et les propriétés physiques qui lui sont propre. Webots est un logiciel de simulation physique de corps solides articulés base sur ODE, un moteur physique. Les résultats de simulation seront confrontés à une série de modèles existants ce qui permettra de valider l'exactitude de la simulation par rapport à la réalité. Une fois ce modèle validé il pourra être un outil de réflexion et de vérification efficace pour l'étude de mouvements complexes.

Le but principal de ce projet est donc de vérifier que Webots est capable de simuler fidèlement un système aussi complexe que les membres inférieurs d'un humain en mouvement et de créer un tel modèle.

Le projet est composé de trois phases distinctes :

1. Modelisation visuelle de l'humanoïde.
2. Mesures des couples aux articulations afin de vérifier l'exactitude de la simulation physique.
3. Commande des articulations en couple (simulant un muscle) et validation de la simulation conjointe de l'orthèse du genou et de l'humanoïde.



(a) Orthèse du genou [LSR] (b) WalkTrainer (LSRO)

FIG. 1.1 – Appareils de rééducation [LSR]

Comme indiqué précédemment, la simulation se concentre sur les membres inférieurs du corps comprenant les muscles et articulations allant du pelvis à la cheville.

La simulation est principalement validée au travers des expériences suivantes :

- la première consistant à contrôler que les couples générés lors de mouvements de marche correspondent à ceux auxquels sont soumis les membres d'un humain.
- La deuxième a été de contrôler les articulations en couples et ainsi de passer à un modèle où les articulations sont modélisées par des servomoteurs à un modèle qui se rapproche plus d'un système musculaire classique.
- Finalement, la troisième et plus importante étape, nécessite l'implémentation d'un modèle de muscle réagissant en fonction d'une stimulation électrique [Mét05] ainsi que la modélisation de l'orthèse du genou développée au LSRO toujours dans le cadre du projet Cyberthèse [LSR]. On contrôle alors que les mesures faites dans la simulation soient similaires aux valeurs obtenues lors d'expérimentations sur des patients réels.

On distingue deux phases principales lors de la modélisation :

La première, environ un tier du projet, est consacrée à l'aspect visuel de l'humanoïde. Il consiste en la création de l'être humain sous Webots en utilisant les tailles des segments humains standard décrites dans le livre [Win90]. Les articulations sont contrôlées au moyen de trajectoires définies par des séries de fourrier provenant d'un modèle Matlab existant. Ce modèle Matlab avait été développé au LSRO par Yves Allamand lors de son travail sur l'orhèse des jambes. Une interface graphique est ajoutée au modèle Webots permettant de mettre en évidence les divers degrés de liberté au moyen de "sliders" influant l'amplitude des mouvements.

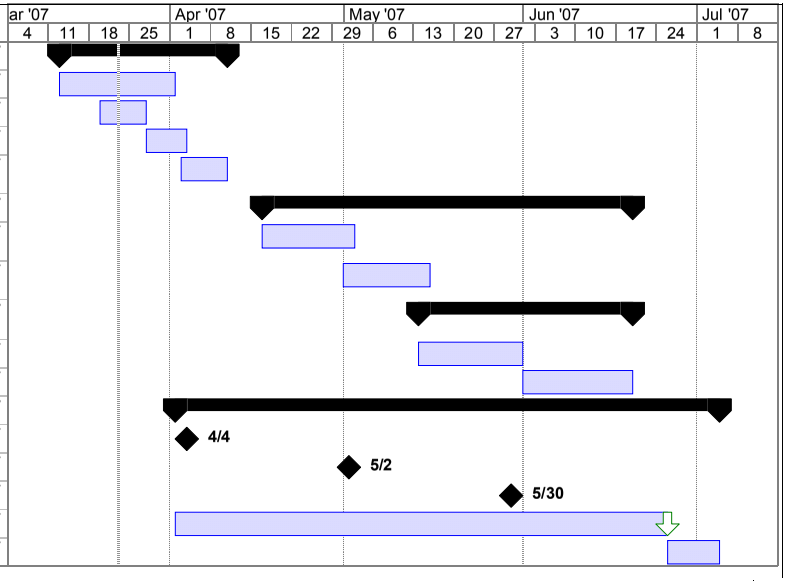
La seconde partie est consacrée à la modélisation physique et à la validation du modèle. Les couples, auxquels les différents segments des jambes et du bassin sont soumis lors de la simulation Webots utilisant les trajectoire définies, ont tout d'abord été mesurés puis comparés avec ceux calculés par Matlab afin

de s'assurer que le modèle répond correctement. Le système de contrôle des articulations, jusque là modélisé par des servos moteur, à été changé pour passer au modèle de muscle développé par P. Métrailler [Mét05] . Les articulations sont donc commandées en couple en fonction de l'angle de la position et de la stimulation électrique auquel le muscle est soumis. L'implémentation laisse la possibilité d'ajouter facilement l'influence de la vitesse en fournissant ce paramètre à la fonction qui calcule la réaction du muscle. L'orthèse du genou a ensuite été modélisée dans Webots et les résultats de la simulation ont été comparés avec les résultats d'expériences réelles.

La suite de ce document se présente de la manière suivante : Le chapitre 2 présente le fonctionnement général de Webots. Dans le chapitre 3, la façon dont est modélisé l'humanoïde est présentée ainsi que l'interface graphique permettant de contrôler l'amplitude des différents degrés de libérté. Le chapitre 4 décris la façon dont sont effectuées les mesure de couples et en présente les résultats. Le Chapitre 5 s'interesse à la façon dont on peut générer un couple sur une articulation dans Webots et présente la façon dont à été implémenté le modèle de muscle. Le chapitre 6 donnes les résultats obtenus à l'aide de la mesure La conclusion (chapitre 8) rappelle les résultats essentiels de ce travail L'environnement de développement comprenant la liste des logiciels et leur version utilisés lors de la phase de création comme lors de la phase de validation sont présenté en fin de document pour permettre la meilleur reproductivitée des résultats possible(chapitre 10).

1.2 Planning original

ID	Task Name	Duration	Start	Finish
1	**Visualisation (30%)**	**22 days**	**Tue 3/13/07**	**Tue 4/10/07**
2	Prise en main du logiciel Webots	15 days	Tue 3/13/07	Sun 4/1/07
3	Création de l'être humain virtuel	6 days	Tue 3/20/07	Tue 3/27/07
4	Contrôler en position les articulations	6 days	Wed 3/28/07	Tue 4/3/07
5	Mise en évidence des mouvements selon chaque degré de liberté	6 days	Tue 4/3/07	Tue 4/10/07
6	**Modélisation (70%)**	**46 days**	**Tue 4/17/07**	**Tue 6/19/07**
7	Commander les articulations en position et mesurer les couples	12 days	Tue 4/17/07	Wed 5/2/07
8	Intégrer le modèle du muscle au niveau de l'articulation	11 days	Tue 5/1/07	Tue 5/15/07
9	**Valider la simulation en créant une orhtèse de genou virtuelle**	**27 days**	**Mon 5/14/07**	**Tue 6/19/07**
10	Modélisation de l'orthèse	14 days	Mon 5/14/07	Thu 5/31/07
11	Validation du modèle	13 days	Fri 6/1/07	Tue 6/19/07
12	**Documentation et Présentations**	**68 days**	**Mon 4/2/07**	**Wed 7/4/07**
13	Présentation Intermédiaire I	0 days	Wed 4/4/07	Wed 4/4/07
14	Présentation Intermédiaire II	0 days	Wed 5/2/07	Wed 5/2/07
15	Présentation Intermédiaire III	0 days	Wed 5/30/07	Wed 5/30/07
16	Rédaction Rapport	61 days	Mon 4/2/07	Mon 6/25/07
17	Présentation Finale	7 days	Tue 6/26/07	Wed 7/4/07



Project: ProjetdeSemestre
Date: Fri 3/23/07

Task — Milestone — External Tasks
Split — Summary — External Milestone
Progress — Project Summary — Deadline

Ce planning à été très bien suivi pour ce qui est de la première partie : Visualisation. Lors de la seconde partie de nombreux problèmes sont apparus lors des mesures de couple et ont retardé la progression du projet. Le retard a été rattrapé dans la mesure du possible sur les parties de modélisation de l'orthèse et de validation du modèle.

Chapitre 2

Introduction à Webots

Ce chapitre est destiné à donner une idée du fonctionnement de Webots au lecteur ne connaissant pas du tout ce logiciel. Il tente de présenter brièvement les bases de son interface et de son fonctionnement.

Webots est un logiciel destiné à la simulation de robots en trois dimensions. Il est développé par la société Cyberbotics en collaboration avec l'EPFL. Le language de Webots est basé sur un sous-groupe du langage de spécification 3D VRML97. Il est conçu pour simuler fidèlement des robots existants ou inexistants, ainsi que le monde qui les entoure. Il permet de créer simplement et rapidement des prototypes, puis de simuler leur comportement. Le portage d'un contrôleur du simulateur au robot réel généralement sans problèmes majeurs et est une des principales qualités de ce programme. Il offre, en plus, l'avantage de pouvoir réaliser des expérimentations en temps accéléré, sans utiliser des robots.

Le logiciel utilise le moteur physique ODE (Open Dynamics Engine) pour gérer la dynamique, les frottements et les contacts. Cette librairie open source est capable de simuler de manière fiable est performante la dynamique de corps rigides.

La documentation de se logiciel se compose principalement de trois documents :
- Le guide utilisateur, [Ltdb] (c'est de ce dernier qu'est tiré la plus grande partie des informations présentées dans ce chapitre)
- Le manuel de référence, [Ltda]
- Le manuel d'utilisateur [Smi06].

2.1 La notion de "monde"

Un monde, dans Webots est un environenement virtuel en trois dimensions dans lequel on peut créer des objets et des robots. Ce monde est donc la description du modèle qui est représenté sous la forme d'un arbre dans lequel chaque objet (robot, sol etc...) est représenté par un noeud et ses fils (qui sont ses paramètres).

2.2 La notion de contrôleur

Un contrôleur est un fichier exécutable qui permet de contrôler le robot. Il accède une API Webots qui est plus ou moins limitée au contrôle des moteurs (c'est ce que l'on cherche à faire dans la majeure partie des cas). Le contrôleur utilisé par la simulation est défini dans le fichier "monde".

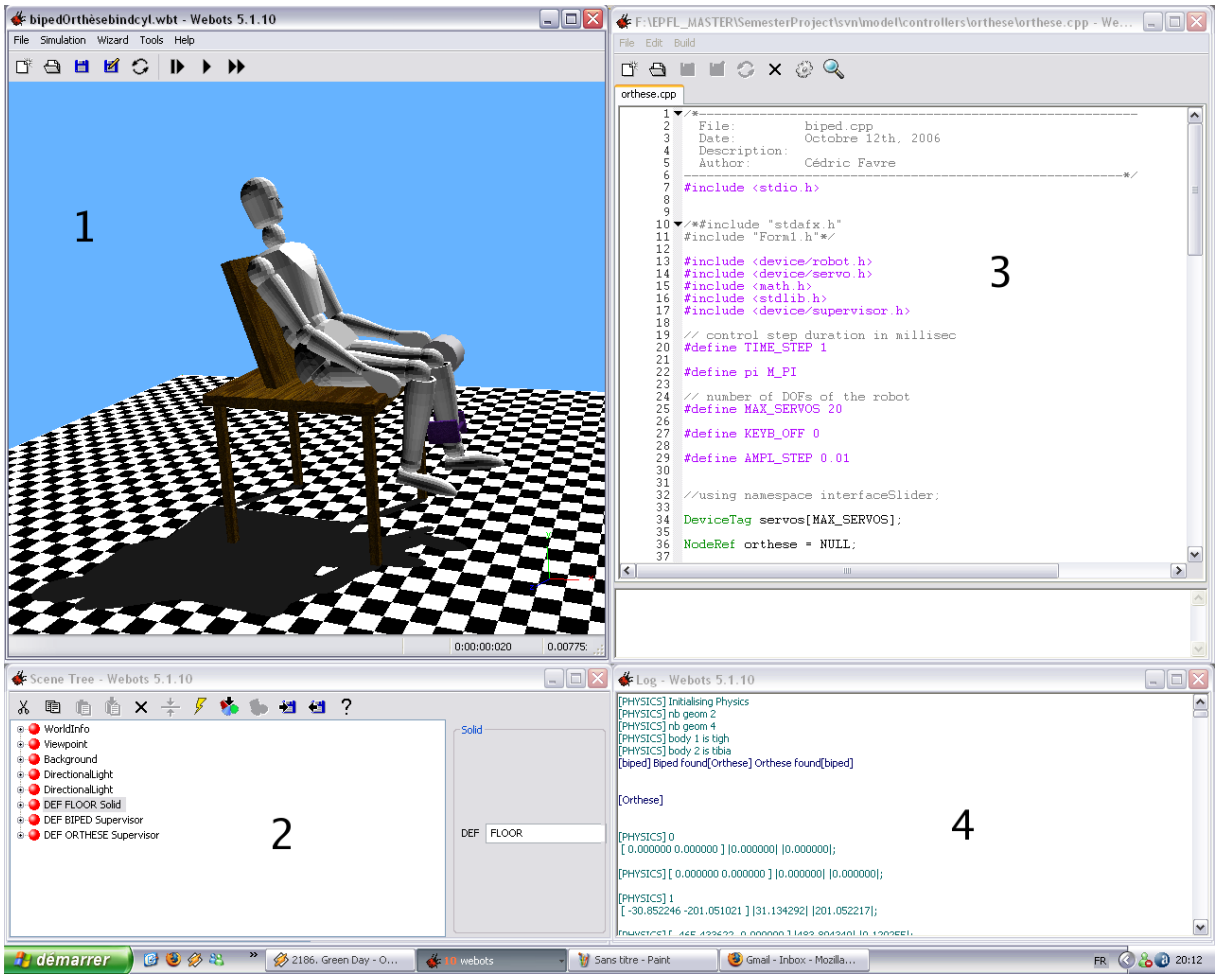


FIG. 2.1 – Interface Webots

2.3 La notion de plugin physique

Un plugin physique est, tout comme un contrôleur, un fichier exécutable. Il permet l'accès direct au moteur physique ODE. C'est de cette manière que l'on peut ajouter des forces s'exercant directement sur les objet du monde ou les lire. Un plugin physique est moins facile à utiliser qu'un contrôleur mais beaucoup plus complet.

2.4 Interface Webots

La figure 2.1 montre l'interface de base du logiciel Webots. Elle possède trois fenètres principales. La fenètre 4 étant un simple log des messages générés par les différents contrôleurs et le plugin physique.

1. La fenêtre de visualisation de la simulation. Elle permet de visualiser ce qui se passe durant les phases de modélisation et de simulation. C'est depuis cette fenêtre qu'on peut ouvrir ou enregistrer les fichiers "monde" [1]. Elle permet aussi de contrôler l'exécution de la simulation (relancer la simulation, avancer, avancer d'un pas de temps, arrêter et avancer en mode accéléré [2].

[1] fichiers avec l'extension .wbt
[2] mode sans affichage visuel qui permet de simuler plus rapidement

2. La fenêtre de modélisation. C'est dans cette dernière qu'on peut modifier le modèle (le fichier "monde"). On peut donc modifier ici les tailles, les paramètres physiques des différents segments, les forces que peuvent exercer les différents moteurs, le contrôleur utilisé par un robot, le pas de temps de la simulation, etc... Les modifications sont appliquées et peuvent être visualisées dans la fenêtre 1.
3. La fenêtre d'édition permet d'éditer et de compiler les contrôleurs et plugins physiques.

2.5 Fonctionnement

Cette section traite de la façon dont les différents composants du logiciel présentés précédemment intéragissent entre eux. La première sous-section expliquent le déroulement d'un pas de simulation. La seconde s'intéresse plus particulièrement à la façon dont Webots commande les servomoteurs, lorsqu'on leur demande d'atteindre une position au prochain pas de temps.

2.5.1 Déroulement de la simulation

Le chef d'orchestre de la simulation est le "timestep", ou pas de temps. Ce pas de temps défini le temps durant lequel la simulation va s'exécuter avant que les prochains calculs (réactions, collisions, contrôle des moteurs etc) s'effectuent à nouveau.

L'explication traduite directement du manuel utilisateur de ODE :

> Le processus de simulation au travers du temps d'un système de corps rigide est appelé intégration. Chaque pas d'intégration fait avancer le temps d'un pas de temps donné, ajustant l'état de tous les corps pour la nouvelle valeur du temps. Il y a deux facteurs principaux à considérer quand on travaille avec un intégrateur :
>
> 1. A quel point est-il précis ? C'est-à-dire, à quel point le comportement du système simulé est en accord avec la réalité ?
> 2. A quel point est-il stable ? C'est-à-dire, est-ce que les erreurs de calculs vont à un moment ou un autre causer un comportement non physique du système simulé ? (par exemple causer "l'explosion" du système sans raison).
>
> L'intégrateur actuel d'ODE est très stable, mais pas particulièrement précis, à moins que le pas de temps soit petit. Pour la plupart des usages faits d'ODE ceci n'est pas un problème. – Le comportement d'ODE parait toujours parfaitement physique dans presque tous les cas. Cependant ODE ne devrait pas être utilisé pour des travaux d'ingénierie quantitatif, à moins que ce problème soit corrigé dans une version future. [Smi06]

Sachant que cette phrase est tirée de la documentation actuelle de ODE et que Webots utilise une version datant de deux ans en arrière, cette explication alarmante donne une forte raison d'être à ce travail pour ce qui est de valider le comportement du modèle d'humanoïde sous Webots.

La suite de ce document se veut heureusement rassurante dans le sens qu'avec un pas de temps assez petit (une milliseconde) on a pu constater des résultats se raprochant de ceux obtenus via générateur de marche Matlab du LSRO ou même des expériences sur des patients réels.

Pour en revenir au procédé de simulation en lui même. A chaque pas de temps Webots appelle le contrôleur pour connaître les "consignes" qui peuvent lui être données (positions à atteindre des moteurs) et applique ainsi certaines forces pour s'approcher de ce but. Si un plugin physique est spécifié il exécute également la méthode `robot_step()` de ce dernier. Puis, il fait évoluer la physique au pas de temps suivant.

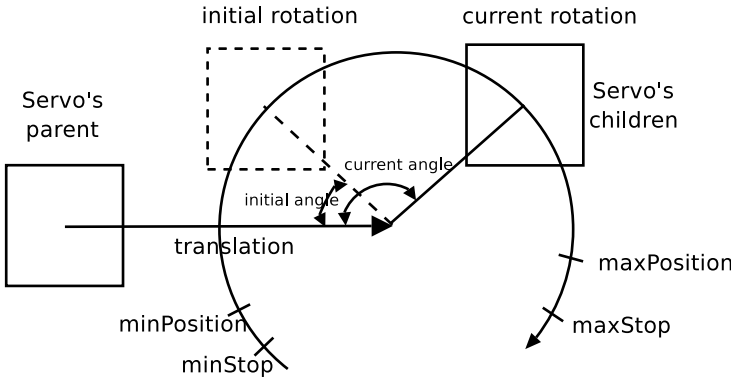


FIG. 2.2 – diagram de servo rotationel (Webots reference manual [Ltda]

2.5.2 Contrôle des servos

Du fait de l'utilisation dans ce projet de servomoteurs comme du fait que comprendre les paramètres les contrôlant a été nécessaire a son bon déroulement. Il apparaît important de présenter la façon dont Webots les modélise et génère les forces (ou plutôt les vitesse) qui les contrôle, ainsi que les paramètres principaux influençant leur réaction[3].

Le fonctionnement des servos est décrit en détail dans la section 2.35 du manuel Webots [Ltdb]. Cette rapide section tente de résumer la partie concernant les servos de type rotationel.

Webots commande les servos en vitesse. Le contrôleur donne généralement la position désirée au prochain pas de temps (même si il y a d'autres moyens de faire). La figure 2.2 montre un sevo rotationel. La procédure suivante présente la manière dont est calculée la vitesse.

entrée :
- P = paramètre de contrôle (donnée par l'utilisateur)
- Pc = position courante
- Pt = position désirée (donnée par le contrôleur)
- Vp = vitesse au pas de temps précédant
- Vd = vitesse maximum que peut atteindre un servo (donnée par l'utilisateur)
- A = accélération maximale dont est capable le servo (donnée par l'utilisateur)

calcule ;
- Vc = vitesse courante
- a = accélération requise pour atteindre Vc
- ts = pas de temps (timestep)

```
Vc = P * (Pt - Pc)
if (Vc > Vd)
    Vc = Vd
if (A != -1){
        a = (Vc - Vp) / ts
        if (a > A)
                a = A
        Vc = Vc + a * ts
}
```

[3]Ils ont du être modifié pour permettre un suivi correcte de trajectoires

La vitesse du servomoteur est donc limitée par quatre facteurs que sont P, Vd, A et une quatrième valeur non présente dans cette procédure : MAX_FORCE [4], qui comme les autres, est spécifiée par le créateur du modèle. Ces quatre paramètres sont donc ceux qui peuvent influencer le comportement du servo et auxquels on fera référence plus tard dans ce rapport.

[4] la force maximal qu'un servo peut exercer

Chapitre 3

Aspect Visuel : Creation d'un humanoïde sous Webots

L'interêt de cette partie du travail est d'obtenir un outil de démonstration pour présenter les mouvements aux patients. Une interface graphique a été réalisée pour mettre en évidence certains de ces mouvements (ce qui est réalisé au moyens de "sliders" contrôlants les amplitudes des différents degrés de liberté du robot).

Le but de ce chapitre est de présenter le travail accompli pour modéliser l'humanoïde à l'aide du logiciel Webots au niveau graphique. Dans cette partie du projet, on ne cherche pas encore un modèle dont le comportement physique est exacte mais plutôt une représentation graphique qui permette de visualiser les mouvements décrits par les différentes parties inférieures du corps durant la marche.

Ce chapitre présente d'abord brièvement la façon dont un modèle est créé. Les données biologiques utilisées sont ensuite détaillées, ainsi que les diverses hypothèses et simplifications du modèle. La façon dont sont générés les mouvements, ainsi que la manière dont ils sont validés sont ensuite expliqués. Le modèle résultant ainsi que son interface graphique sont ensuite commentés.

3.1 Modélisation

3.1.1 Convention pour le sens des axes

La figure 3.1 présente la manière dont sont posés les axes x y et z pour le reste des discussions au sujet de ce projet. C'est la convention proposée par "The Standardization and Terminology Committee (STC) of the International Society of Biomechanics (ISB)" [SotISoBI02] et accessoirement aussi par Webots.

3.1.2 Manière de Modéliser

Pour ce projet, l'humanoïde est donc modélisé par un robot dont les caractéristiques physique des segments sont fournies par des mesures biologiques. Dans cette première partie de projet, les articulations sont contrôlées par des servomoteurs qui n'ont nullement la prétention d'être la simulation adéquate d'un muscle mais dont la simple tâche consistera à suivre des trajectoires données. Lors de la deuxième partie du projet, ces servomoteurs sont supprimés et remplacés par un modèle de muscle.

La modélisation s'est déroulée en trois étapes principales :

1. En premier lieu, la géométrie du robot est créé, ce qui comprend les différents segments, leur taille et les joints qui sont la manière de les fixer l'un à l'autre. La figure 3.2(a) présente le robot au terme de cette étape, les formes visibles sont présentes pour aider à la visualisation et seront supprimées pour l'étape suivante d'enrobage.

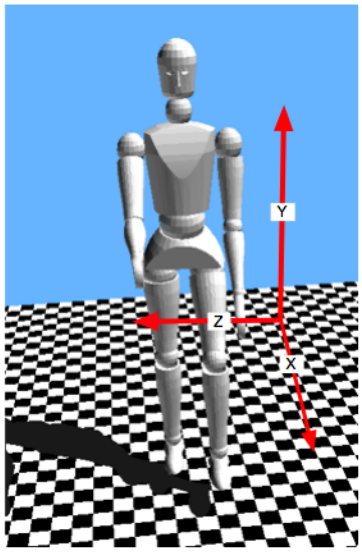


FIG. 3.1 – Systeme d'axe selon [SotISoBI02]

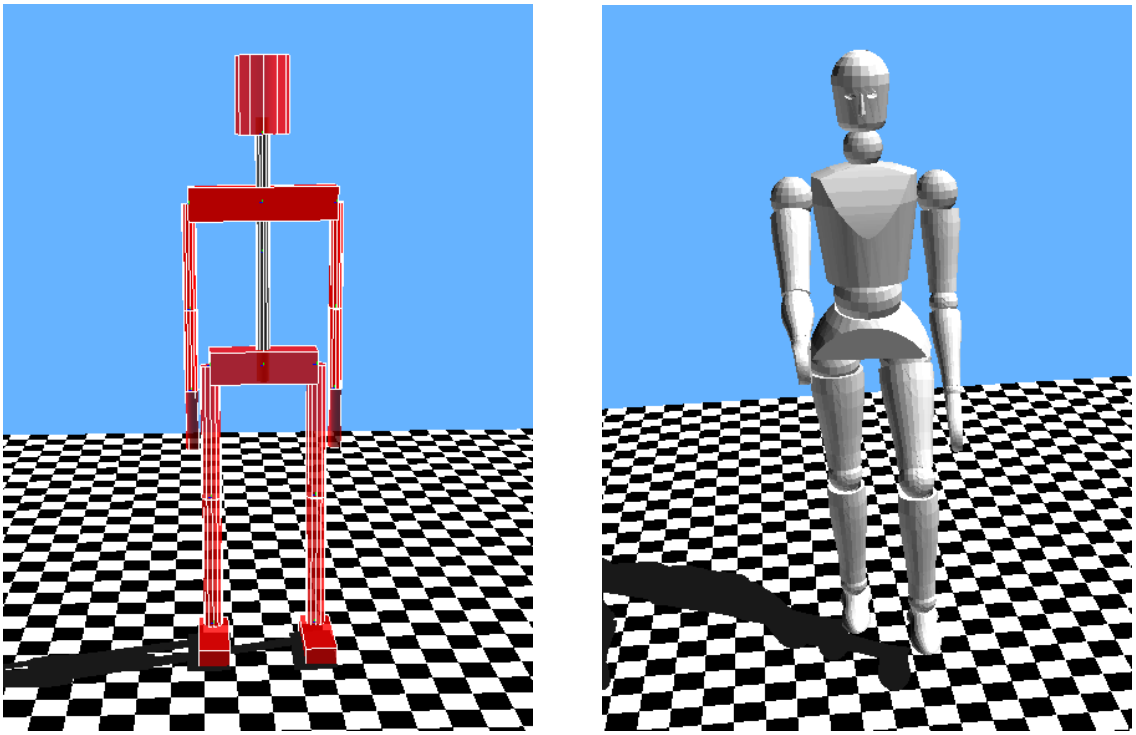
2. Dans un deuxième temps, on ajoute l'enrobage VRML qui avait été dessiné pour un autre projet dans le logiciel pro-egineer. Cet enrobage doit être importé puis appliqué sur les différentes parties du corps. La figure 3.2(b) présente le résultat une fois cette étape effectuée.
3. Les grandeurs physiques tels que le poids, la force des moteurs et la position des centres de gravité des différents segments sont alors ajoutés et adaptés au modèle.

Les étapes 2 et 3 ne sont pas liées et pourraient tout à fait être effectuées dans un autre ordre. L' une d'elle pourrait même ne pas être effectué selon si on ne se préoccupe pas de l'exactitude physique ou de l'aspect du modèle. Dans le cadre de ce travail elles ont été réalisées de manière relativement parallèle puisque la physique à été testée avant d'ajouter la texture et ajustée à plusieurs reprises par la suite.

3.1.3 Données Biologiques et Hypothèses

Comme mentionné plus haut, dans cette partie on recherche un modèle le plus exact possible du point de vue des mouvements et des divers aspects visuels. Les grandeurs physiques ne sont par contre pas prioritaires. Ce modèle comporte un certain nombre d'hypothèses dont les principales sont listées ci-dessous. La source principale des données biologique est le livre "BIOMECHANISCS AND MOTOR CONTROL OF HUMAN MOVEMENT" de David A. Winter [Win90].

1. Le squelette du robot est crée selon la figure 3.3
2. Les tailles des différents segments proviennent du livre [Win90] elles sont calculées pour un humain mesurant 175 centimètres. Le calcul se base sur les données de la figure 3.3.
3. Les masses des différents ségments sont présentés dans la table 11.3. Ces données ont été calculées à partir de la même source que celles du points précèdent.
4. Les trajectoires permettant de générer le mouvement de marche, proviennent du modèle Matlab du LSRO sous forme de séries de fourrier et sont présentées dans la section suivante (voir 3.2).
5. La position des jambes a été rapprochée pour concorder avec l'aspect graphique fourni et ainsi obtenir un humanoïde plus réussi. En effet alors que sur le modèle du squelette les jambes sont placées aux extrémités du bassin, ce n'est pas le cas de l'enrobage. Cette hypothèse est faite car il parait plus important de conserver un rendu graphique plaisant que d'être exacte au niveau de l'espacement des jambes. Ce dernier n'a aucune influence dans le cadre des expériences menées



(a) Modèle sans texture (b) Modèle avec texture

FIG. 3.2 – Différentes phases de modélisation

dans ce projet. Cependant il pourrait s'avérer important de corriger cet espacement pour d'autres projets comme, par exemple, si l'on en venait a simuler le "WalkTrainer".

6. Encore une fois pour des raisons graphiques et d'animation, l'articulation entre le bassin et la colonne vertébrale a été déplacée contre le haut, par rapport au squelette original. Ceci pour concorder avec l'enrobage VRML fourni. Cette hypothèse encore a, à nouveau, été faite pour améliorer le résultat graphique. De plus, dans le cadre des expériences réalisées, cette modification ne semble pas avoir une importance capitale.
7. Les mouvements des bras sont tout à fait artificiels et sont générés en utilisant les trajectoires des jambes un peut décalées pour paraître crédibles. Ces trajectoire ne sont donc pas les trajectoires des bras d'un humain en train de marcher.
8. Les centres de gravité des jambes sont calculés à partir de [Win90].
9. Les segments sont représentés physiquement par des cylindres dont la section ne provient d'aucun modèle. Elle est adaptée pour correspondre à l'enrobage VRML fourni.
10. Le déphasage en de l'articulation des hanches avec la cuisse à du être adapté. En effet le modèle matlab fourni ne semble pas tenir compte des déphasages induits par les articulations supérieures. On peut constater en utilisant les équation de mouvement d'origine du modèle Matlab que par l'inclinaison du bassin a pour effet de décaler les jambes vers l'arrière. Ceci a été corrigé, en compensant cette inclinaison au niveau de l'articulation de la hanche, pour replacer les jambes droites.

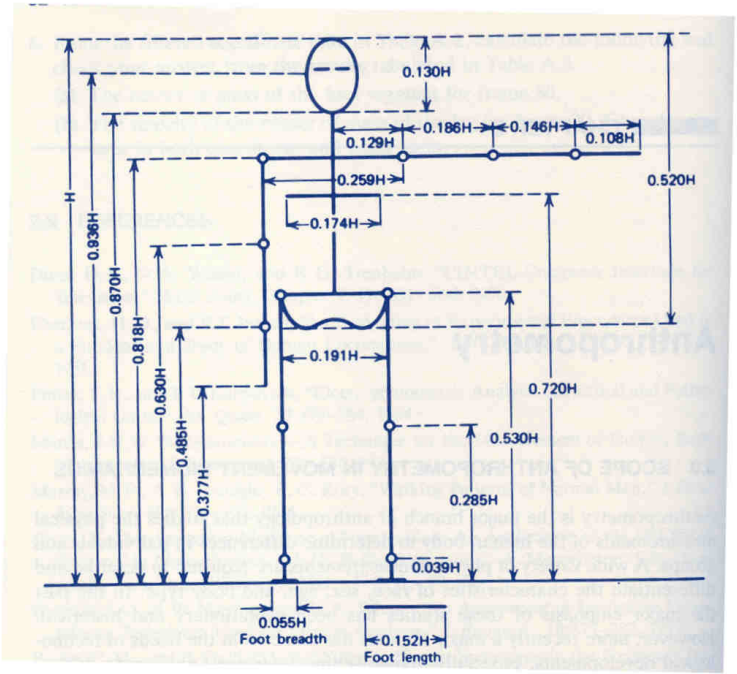


FIG. 3.3 – Tailles des segments [Win90]

3.2 Mouvements

3.2.1 Génération des trajectoires

Afin d'obtenir un mouvement de marche, on modélise les articulations du robots par des servomoteurs que l'on commande à l'aide de trajectoires tirées d'un modèle Matlab développé par le LSRO. Les trajectoires sont données par des séries de fourrier du 4ème ordre donnant l'angle de l'articulation en fonction du temps (voir Fig. 3.4). Les paramètres relatifs à chaque articulations sont données dans la table 11 en annexe.

$$\begin{aligned}\Theta &= a_0 \\ &+ a_1 * cos(w*t) + b_1 * sin(w*t) \\ &+ a_2 * cos(2*w*t) + b_2 * sin(2*w*t) \\ &+ a_3 * cos(3*w*t) + b_3 * sin(3*w*t) \\ &+ a_4 * cos(4*w*t) + b_4 * sin(4*w*t)\end{aligned}$$

FIG. 3.4 – Série de Fourrier du 4^{me} ordre

3.2.2 Suivi des trajectoires

Dans le but de s'assurer que la position des jambes est bien celle que l'on a commandé. On compare la position désirée avec la position effective des servos.

Résultats

La figure 3.5 montre que les trajectoires données en paramètres sont bien suivies pour les 3 articulations. La courbe en traitillé représente la position désirée (en radians) alors que la ligne pleine décris la position au temps donné du servo.

Les paramètre associés aux servos pour atteindre ces résultats sont présentés dans la table 3.1. La façon dont ces paramètres influencent le contrôle de la trajectoire avait été détaillée dans le dans la section 2.5.2. La section suivante revient sur ces valeurs et leur influence.

max force [N]	max speed [m/s]	controlP	accélération [m/s^2]
5000	5000	100	-1

TAB. 3.1 – Paramètres des servos

Importance des divers paramètres du moteur

Pour obtenir un tel suivi les 4 paramètres présentés dans le tableau 3.1 doivent être adapté. Max force, accélération et max speed ne sont que des bornes qui représentent la force, l'accélèreation et la vitesse maximal que le servo peut atteindre . Dans notre cas ils sont réglés de manière assez large [1].

Le paramètre P est le paramètre le plus critique et le plus complexe à déterminer.

La figure 3.6(a) montre comment le postion évolue avec la valeur P par défaut de Webots. On constate que le servo n'arrive pas à suivre sa trajectoire. (La valeur par défaut étant petite due au fait que généralement les modèles Webots sont créés pour des robots dont les tailles et masses sont généralement plus petites que celle de notre humanoïde)

La figure 3.6(b) montre le suivi du servo de la hanche dans le un cas où P a été réglé à une valeur trop grande. On remarque alors que la trajectoire et suivi mais que le servo ne se stabilise pas à la la bonne position.

3.3 Interface graphique

Cette section présente le modèle humanoïde accompagné de son interface graphique. ([2], [3])
Sur la vidéo jointe à ce rapport on peut voir

3.3.1 Résultat Visuel

La figure 3.7 présente l'interface graphique permettant de contrôler les articulations et ainsi de mettre en évidence les divers mouvements. Les sliders représentent l'amplitude des trajectoires aux différentes articulations.

3.3.2 Adaptation dynamiques des amplitudes

L'amplitude des mouvements au niveau de la hanche autour des axes x et y ne sont pas modifiables directement depuis l'interface graphique.

Ceci pour trois raisons :

1. On essaie de garder l'interface la plus simple possible pour l'utilisateur.

[1] une valeur de -1 pour l'accélération et une façon de dire a Webots que l'accélération peut être infinie

[2] Cette interface graphique fonctionne et a été testé uniquement pour le système d'exploitation Windows XP. Pour une utilisation Mac OS ou Linux elle devrait être adaptée aux librairies graphique et système de threading du système d'exploitation et recompilé.

[3] Ayant constaté à plusieurs reprises que Webots n'arrivait pas toujours à lancer l'interface si elle était stockée sur un volume de stockage externe tel une clé usb ou un lecteur réseau. Il est conseillé de copier sur le disque dur et de l'exécuter depuis là.

2. Ces mouvements ne font pas parties des mouvements que l'on cherche à mettre en évidence. Ils sont uniquement présents dans le modèle pour garder une démarche réaliste.
3. Ils sont fortement couplés aux amplitudes colonne-hanche et les modifier indépendamment résulterai en un mouvement faux..

Il sont donc calculés directement à partir des amplitudes colonne-hanche. En effet, en l'absence de ces mouvements ou si ils sont mal adaptés, le mouvement du bassin et le fait que notre humanoïde est suspendu, impliqueraient que les jambes se balancent de gauche à droite du fait de la rotation du bassin autour de l'axe x. De plus, on observerait une rotation des jambes autour de l'axe y induit par la rotation du bassin autour de ce même axe.

3.4 Programme de génération de modèles

Dans le but de rendre la modification de certains paramètres des modèles Webots plus rapide, un petit logiciel Java a été implémenté. En effet, devoir modifier certains paramètres depuis l'interface du logiciel, comme par exemple la force ou la vitesse maximale des moteurs, peut s'avérer fastidieuse.

Les fichiers "monde" de Webots portants l'extension ".wbt" sont de simples fichiers texte. Le programme prend donc en entrée un "template", qui n'est autre qu'un fichier monde de Webots [4] auquel a été ajouté certaines balises. Il le modifie pour ressortir un fichier ".wbt" à.

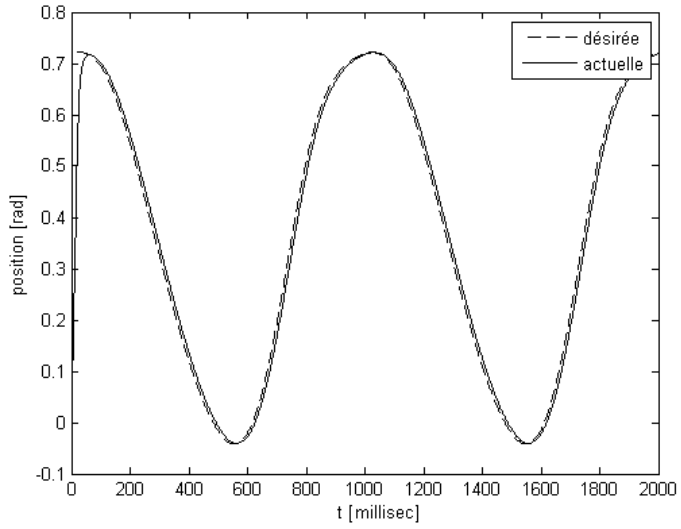
Le programme cherche dans le fichier tpl des balises qu'il remplacera par la valeur désirée. Une balise, telle que définie actuellement, se présente sous le format <@NOM_BALISE>

Le programme actuellement modifie pour chaque moteur les trois paramètres suivants :
- max force : la force maximale qu'un moteur peut exercer.
- max speed : la vitesse maximale à laquelle un moteur peut opérer.
- controlP : le paramètre P qui modifie la vitesse réaction du moteur. (comme expliqué dans la section 2.5.2)

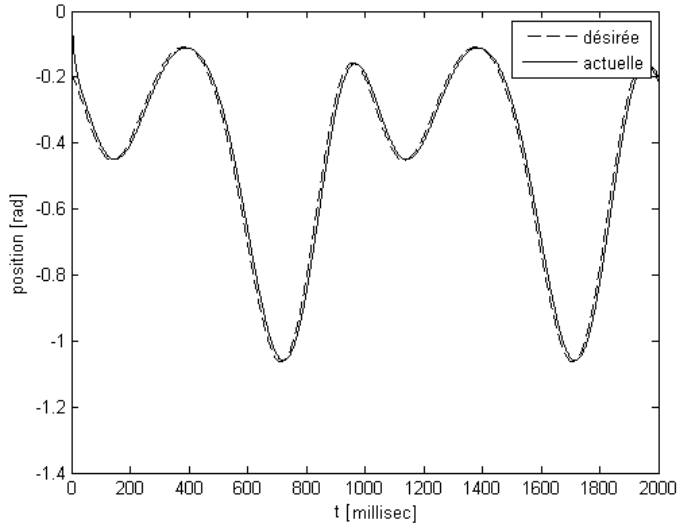
Ces balises sont facilement augmentables. Il suffirait d'ajouter dans le code la balise et la valeur par laquelle on aimerait la remplacer (tout en placent la balise dans le fichier template). On pourrait aussi imaginer définir les balises dans un fichier séparé pour ne pas avoir à modifier le code. La valeur par laquelle les différentes balises pourrait aussi être calculée par le programme ainsi.

Ces améliorations peuvent aller plus loin et apporter la possibilité d'avoir un modèle dont les différents paramètres (tailles, masses, centres de gravitée) seraient variables. On peut donc imaginer étendre ce programme pour lui permettre de générer des modèles physiologiquement différent tel un petit gros ou un grand maigre. Les enrobages doivent aussi être adaptées aux nouveaux paramètres. Il semble réaliste de s'adapter à la taille (il existe un champs "scale" qui permet d'adapter la taille de chaque partie de l'enrobage). La masse parait beaucoup lus difficile à refléter sur le modèle (le petit gros aura difficilement l'aire gros).

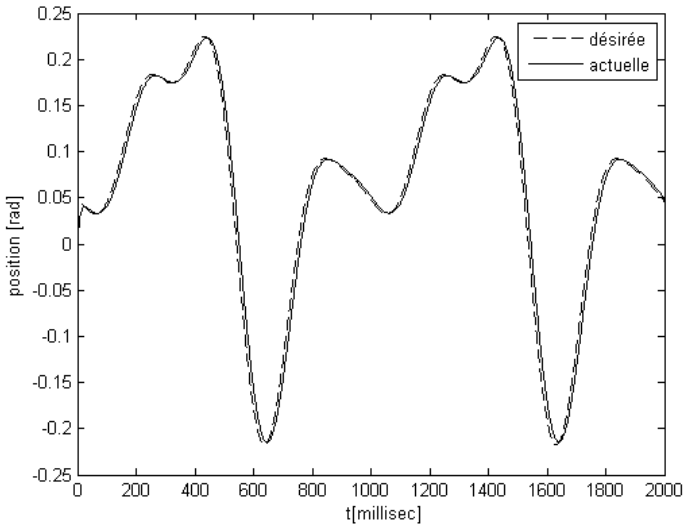
[4] un fichier ".wbt" dont l'extension à été changée en ".tpl"



(a) Hanche

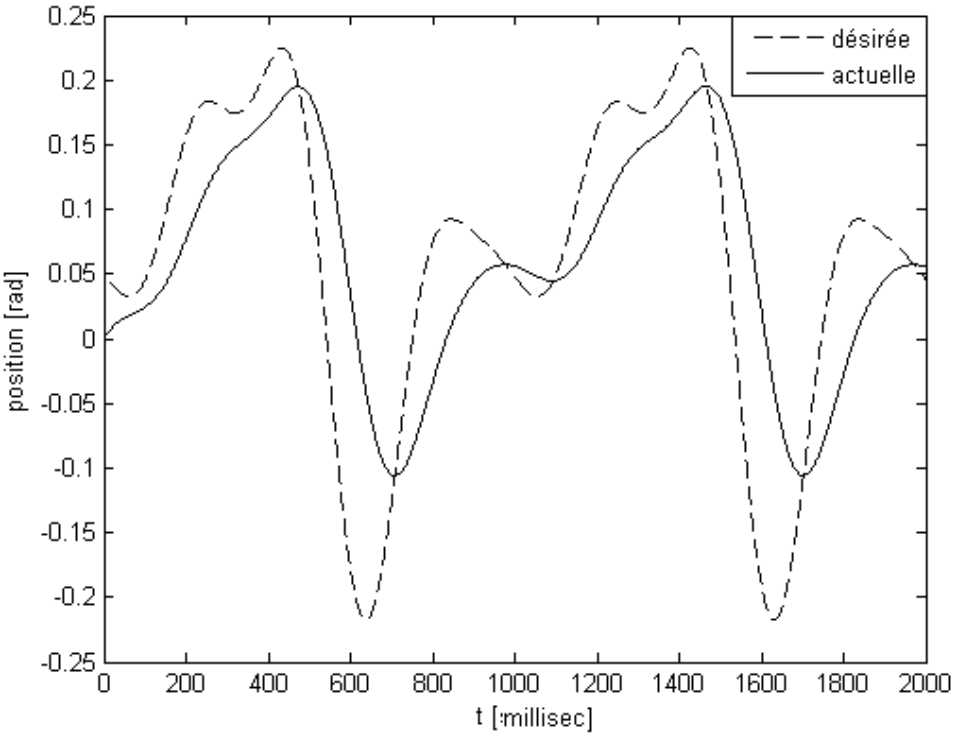


(b) Genou

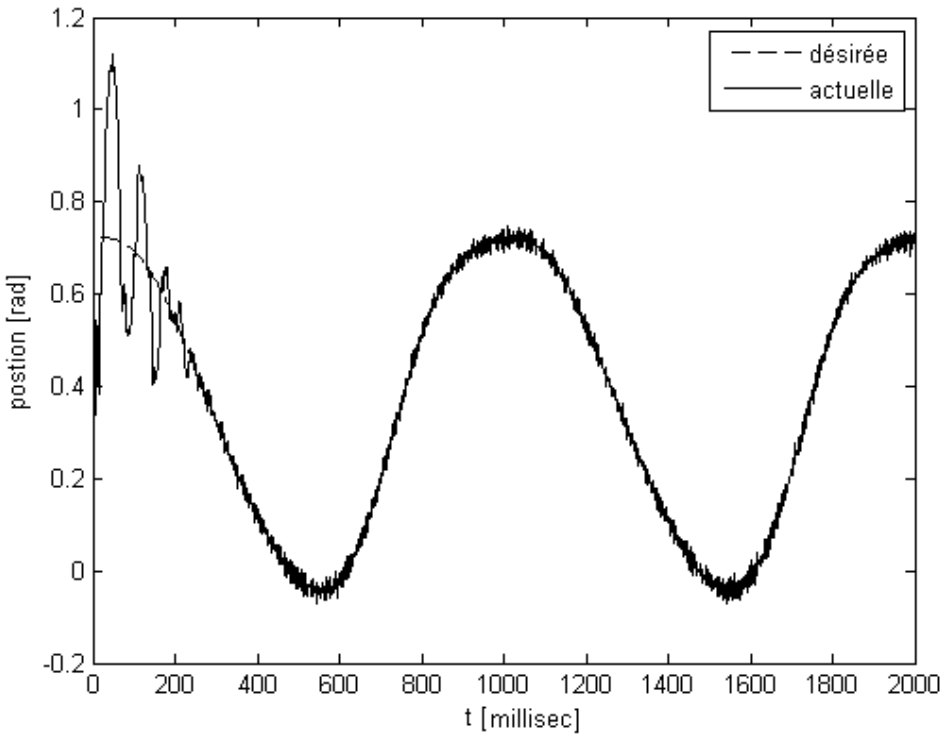


(c) Cheville

FIG. 3.5 – Suivi des positions



(a) P = 10



(b) P = 3000

FIG. 3.6 – Examples de mauvais suivi de positions dus à un P inadapté

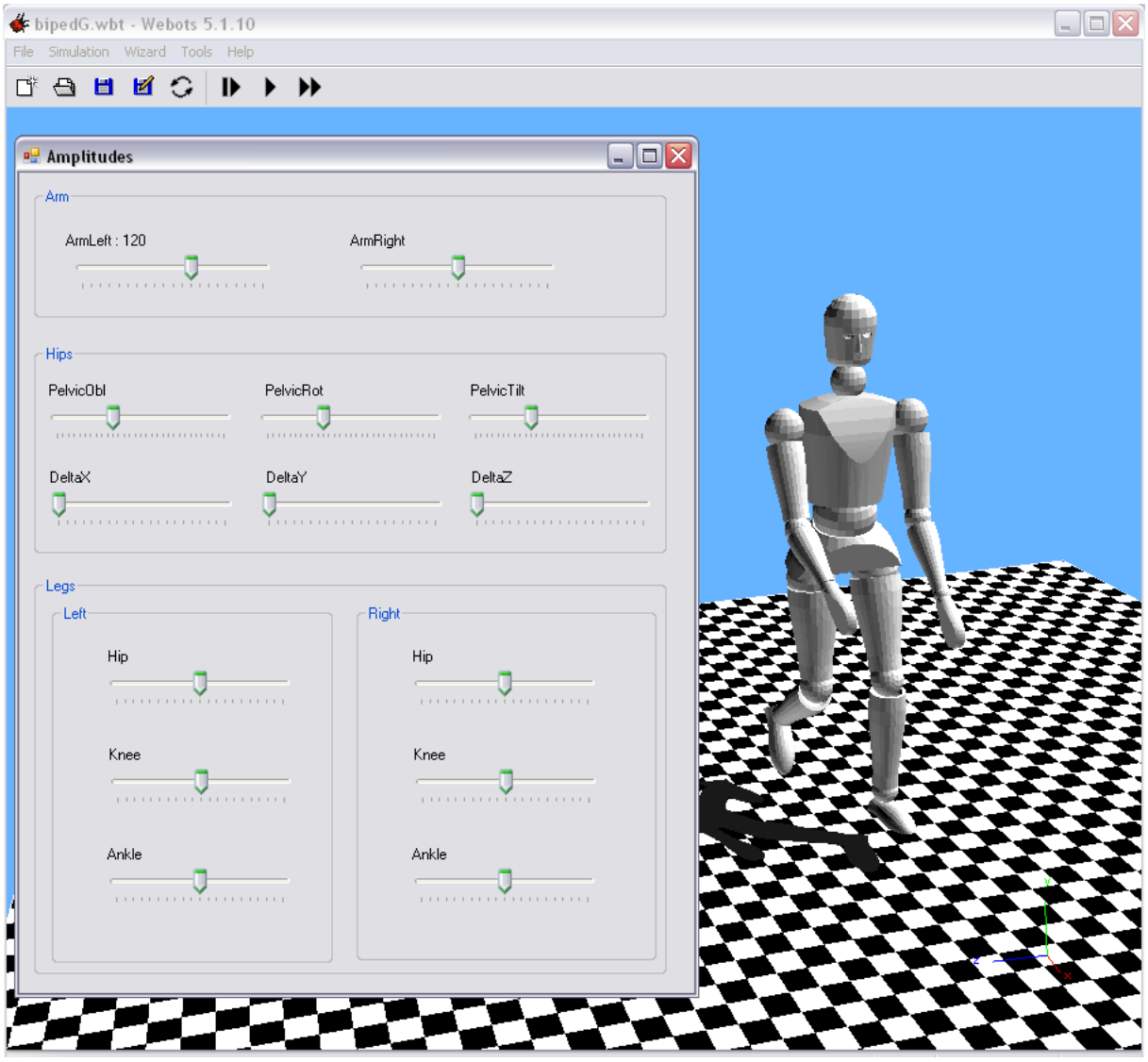


FIG. 3.7 – Biped avec Interface Graphique

Chapitre 4

Mesure des couples lors de la marche

4.1 But

Afin de valider le modèle Webots par rapport au modèle Matlab il était important de mesurer les couples obtenus lors des mouvements induits par les trajectoires présentées à la section 3.2. Ceci a été réalisé par l'intermédiaire d'un plugin physique qui lit les couples aux articulations appliquées par Webots afin d'atteindre les positions commandées. Ceci permet de s'assurer que la physique appliquée par Webots est similaire à celle imaginée dans le modèle Matlab.

4.2 Problèmes

Lors des mesures, plusieurs problèmes sont survenus. Ils sont brièvement décrit dans cette section de sorte qu'une personne désireuse de réutiliser ce travail soit enclin à le faire dans les meilleures conditions possibles.

- Fonctions non mise à jour correctement : certaines fonctions d'ODE ne sont pas appelées au moment adéquat par la boucle principale de Webots pour pouvoir être lues dans le plugin physique. Ceci est le cas de la fonction `dBodyGetTorque (dBodyID)` et probablement de la fonction parallèle `dBodyGetForce (dBodyID)`. Lors des mesures suivantes la structure Feedback, qui contient les couples et forces appliqués aux deux corps attaché à un joint, a été utilisée et a finalement permi d'obtenir les valeurs désirées.
- Dans la dernière version de Webots (5.12 actuellement), un bug dans l'appel du contrôleur a pour effet que ce dernier est appelé une fois sur deux dans le cas d'exécution avec un pas de temps petit (de l'ordre d'une miliseconde). Ce phénomène induit de gros problèmes lorsque l'on s'intéresse à la valeur des couples.
  On remarque que du point de vue de la physique la position désirée ne change qu'une fois sur deux, ceci implique qu'un grand couple doit être appliqué pour atteindre la position au pas de temps suivant, puis on doit fournir en grand couple dans le sens opposé pour permettre de garder l'articulation à cette position pour le prochain pas de temps. Au prochain pas de temps la position sera mise à jour et donc on devra à nouveau appliquer un couple dans l'autre sens pour atteindre cette position et ainsi de suite... Ceci implique de très grosses oscillations au niveau des couples mesurés et donc pratiquement des mesures fausses et inutilisables.
  Pour cette raison ce travail a été réalisé en utilisant la version 5.10 de Webots qui ne présente pas ce bug. Si les prochaines version n'étaient pas corrigées il faudra continuer à utiliser cette version. [1]
- La valeur que retourne Feedback n'est pas documenté. Au vu des problèmes rencontrés dans son utilisation pour les mesures de couples tenant compte de l'électro-stimulation et des tests ensuite réalisés, nous avons conclu que cette structure retournait les couples et forces calculés par ODE,

[1] Lors d'une discussion récente avec le programmeur principal de Webots, il m'a indiqué qu'il était en train de travailler à la correction de bug

mais ne prenait pas en compte les forces ajoutées manuellement. Sachant cela, cette structure retourne les résultats désirés de manière fiable, il est donc très important d'être au courant de ce fait qui a été à l'origine de bien des maux de tête.
- Pour que les mesures de couple soient correctes, la façon de fixer le robot dans les airs est importante. Jusqu'ici il était maintenu en le replaçant à une position donnée à chaque pas de simulation (ce qui permettait aussi de déplacer le robot en accord avec les translations du bassin). Ceci fonctionne très bien d'un point de vue visuel. Cependant, il s'est avéré lors des mesures que les forces créées lors du pas de temps ne sont pas toutes remises à zéro et il en résulte des mesures erronées. La meilleure solution testée consiste à créer dans le plugin physique de Webots un joint fixe entre le robot et le "monde" qui bloque sa position comme si ce dernier était physiquement suspendu.

4.3 Conditions de mesure

Du fait qu'il est extrêmement difficile de faire marcher un robot humanoïde de manière stable et que ce n'est pas le but de ce travail, les mesures ont été effectuées sur un modèle fixé dans les airs. Les forces provenants des réactions avec le sols ne sont donc pas présente bien que ces dernières changent radicalement la forme et l'amplitude des couples qui seraient mesurés. Par chance, le modèle Matlab qui calcule les couples théoriques auxquels nous avons comparés les résultats, est facilement paramêtrable pour ne pas tenir compte des ces interactions. Les résultats présentés ci-dessous semblent donc indiquer une très bonne correspondance entre les valeurs mesurées dans la simulation Webots et celles obtenues théoriquement par Matlab. Cependant, de part la simplification faite, ils ne sont en aucun cas les mêmes que les couples réels exercés lors de la marche.

4.4 Résultats

La figure 4.1 présente les résultats des mesures superposés aux valeurs générées par le modèle Matlab.

Il y a deux constations principales :

1. La forme des courbes est tout à fait équivalente. Ce qui montre que Webots simule de façon fidèle les différentes interactions entre les différents segments en mouvement.
2. L'amplitude des courbe présente des différences significatives.
   Quelques constatations :
   - Les résultats pour la hanche ne présentent que des différences minimes.
   - Les résultats pour la chevilles sont clairement les plus éloignés du modèle Matlab.
   - L'erreur semble augmenter au fur et à mesure que l'on descend le long du corps.

   Explications possibles :
   (a) Le modèle physiologique du modèle matlab est différent. (tailles, masse, centre de gravitée)
   (b) L'erreur incrémentale
   (c) Les offsets des différents segments.

Pour tenter d'y voir plus claire, la figure 4.2 montre les résultats pour des mesures effectuées sur la cheville isolée (c'est-à-dire : seule la cheville bouge).

On remarque que :

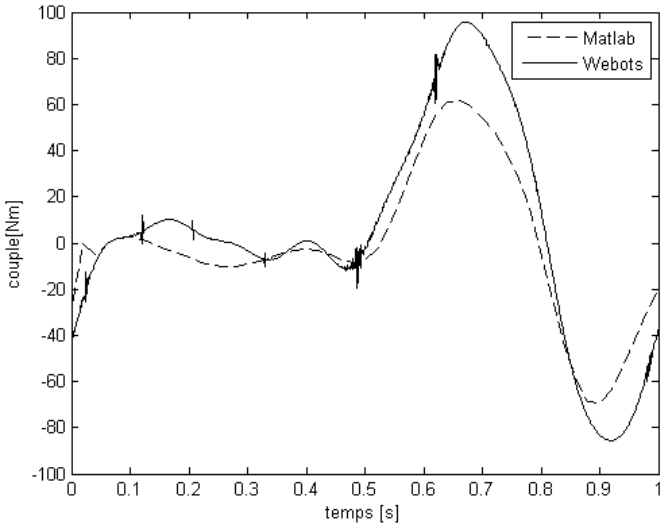
1. La courbe est toujours bien suivie et l'amplitude toujours différente.
2. L'erreur semble sensiblement plus faible.
3. Il y a un décalage (les deux courbes ne sont pas centrées sur le même axe).

Cette nouvelle mesure semble confirmer l'hypothèse de l'erreur incrémentale. Le décalage complet de la courbe peut venir de l'offset. Les différence semblent toujours très grandes, comparons les deux modèles :

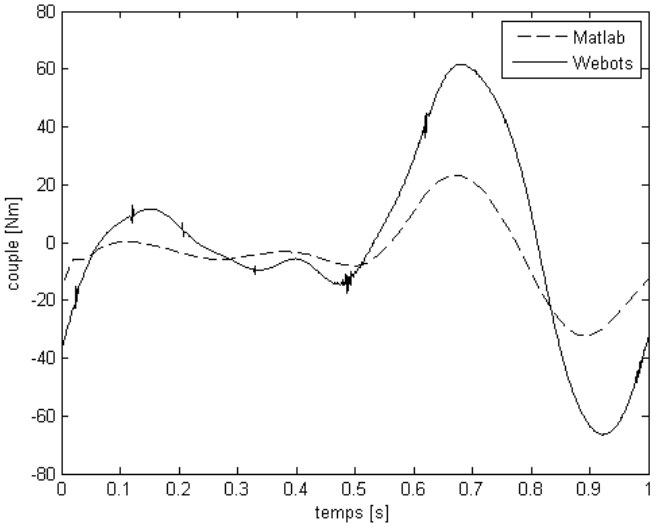
Malgré les explications énoncées, ces erreurs paraissent très grandes (surtout pour la cheville). Il serait donc bon d'investiguer plus en détails les hypothèse mentionnées ci-dessous. Celà n'a pas été fait pour les raisons suivantes :

- Le modèle matlab devrait être fortement changé pour prendre en compte exactement les même données biologiques.
- Le problèmes est apparu assez tard dans le projet. En effet, les courbes obtenues originalement étaient plus vraisemblables car les articulations ne suivaient pas correctement la commande (du a un paramètre P trop faible).
- Du à l'apparition tardive de ces erreurs trop grandes, du fait qu'il était plus interressant de passer à la suite et que l'on utilise plus ces servos par la suite (on commande les articulations en couple). J'ai jugé préférable d'avancer la suite du projet.

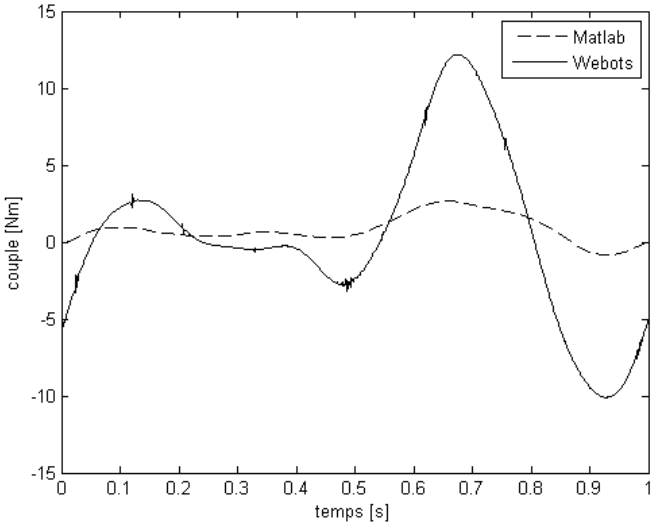
Pour résumer ce chapitre de mesures on peut donc constater que pour ce qui est de la forme les couples mesuré sont tout à fait conforme au modèle Matlab. Fait normal sachant que les mêmes trajectoires sont utilisées par les deux modèles. Par contre l'intensité des couples diffère, ce qui semble dû aux différences de modèles utilisés.



(a) Hanche



(b) Genou



(c) Cheville

FIG. 4.1 – Couples mesurés aux différentes articulations

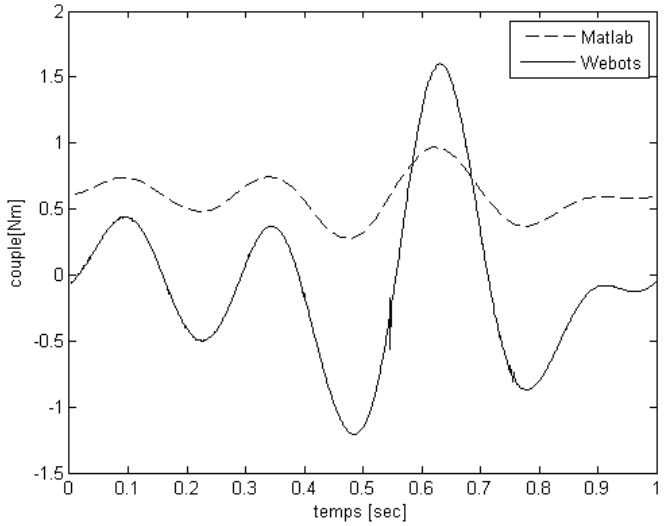


FIG. 4.2 – couples générés sur la cheville isolée

Chapitre 5

Commande des articualtions en couple

Il s'agissait ici de controler les articulations en couple pour modéliser l'influence d'un muscle électro-stimulé. Un premier test a été effectué à l'aide d'un contrôleur de type PID extrêmement simplifié. Un modèle de muscle électro-stimulé a ensuite été implémenté.

L'orthèse du genou a ensuite été modélisé. La partie résultats présente la phase de validation qui cherche à démontrer que le modèle réalisé sous Webots est conforme à la réalité et simule de manière représentative les interactions entre l'humanoïde avec son modèle de muscle électro-stimulé et l'orthèse du genou.

5.1 PID

5.1.1 Définition

Un PID (Proportional-Integral-Derivative) contrôleur est un moyen de commander une articulation en force ou couple par rapport à une position désirée. La réponse en force ou couple sur les corps rattachés au "moteur" est commandée par l'erreur e entre la position désirée et la position actuelle. Il est composé de trois parties :

1. La partie proportionnelle : calcule une contribution par rapport à l'erreur actuelle.
2. La partie intégrale : génère une contribution par rapport au passé récent de l'erreur.
3. La partie dérivée : fournit une contribution par rapport à la vitesse de l'évolution de l'erreur.

L'équation 5.1 donne la façon dont est généralement calculée la réaction (en couple ou force d'une articulation au temps t par rapport à une erreur e. Elle est représentée d'un moyen plus graphique par la figure 5.1.

$$Output(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de}{dt} \tag{5.1}$$

Les constantes K_p, K_i et K_d doivent être adaptées selon la dynamique du problème pour obtenir le mouvement désiré sans oscillations.

5.1.2 Implémentation

Dans le cadre de ce projet c'est uniquement la partie proportionnelle qui a été implémentée et testée. Le but ici n'était pas d'avoir un contrôleur PID parfait, mais simplement de se familiariser avec la physique du simulateur (ODE) et ainsi d'écrire un premier plugin physique fonctionnel qui permette de commander l'articulation en couple. Ceci a permi, outre l'apprentissage du fonctionnement de la librairie physique, de mieux comprendre comment passer d'un contrôle en position à un contrôle en intensité (couple ou force) qui peut être utilisé dans le cas de la simulation comme dans le cas de robots réels.

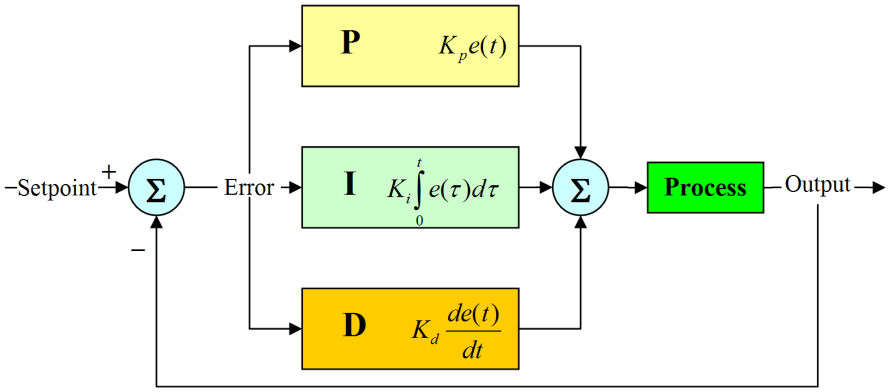


FIG. 5.1 – diagramme PID (Wikipedia)

5.2 Integration du modèle de muscle

Le modèle de muscle utilisé est celui présenté par monsieur Patrick Métrailler dans sa thèse [Mét05]. Il comporte certains paramètres propres à chaque muscle, qui déterminent sa réponse en fonction de l'angle de l'articulation et de la stimulation électrique appliquée. Il comporte aussi d'autres paramètres décrivants les forces passives exercées par le muscle, les ligaments et autres forces bio-mécaniques présentes chez un humain. Dans le cas de stimulation réelles, tous ces paramètres sont déterminés dans une phase d'identification réalisée directement sur le patient.

Ce modèle doit être écrit dans la physique même du simulateur et on utilisera la possibilité offert par Webots d'écrire un plugin physique. Ce dernier permet d'accéder aux fonctions de la librairie physique ODE et ainsi d'appliquer directement des couples et des forces sur les objets modélisé.

5.2.1 Modèle

Ce modèle prend en considération la position en degré de l'articulation et l'intensité de la stimulation. Il ne dépends pas d'autres facteurs comme la fréquence de la stimulation, la fatigue la vitesse ou autre. Dans le but d'être facilement améliorée l'implémentation actuelle fourni un paramètre $\dot{\Theta}$ qui permettra d'ajouter facilement l'influence de la vitesse dans le calcul de la réaction musculaire. Une vision schématique de ce modèle est donnée par la figure 5.2.

L'équation Fig 5.3 présente la manière dont est calculé le Moment musculaire créé par un muscle, elle ne prend cependant pas en compte le moment passif qui, dans le modèle de P.Métrailler, est lui aussi exprimé par un polynôme de degré trois, tout comme $\bar{M}$. On s'attend à ce que Webots simule uniquement le moment passif généré par la gravité. Les autres forces ne sont pas inclues dans cette version du modèle.

Une interpolation linéaire du 1^{er} ordre, donnée par l'équation Fig. 5.4, est réalisée pour approcher au mieux la valeur de la réaction du muscle à la stimulation (d'après la formule de Taylor-Young au premier ordre). Cette formule est évidemment adaptée pour les caculs aux extrémités (c.à.d. quand on ne connais pas de I^+ ou I^-). Dans ce cas on retourne la valeur la plus proche connue (car la courbe de réaction au courant est plate dans les cas extrêmes) .

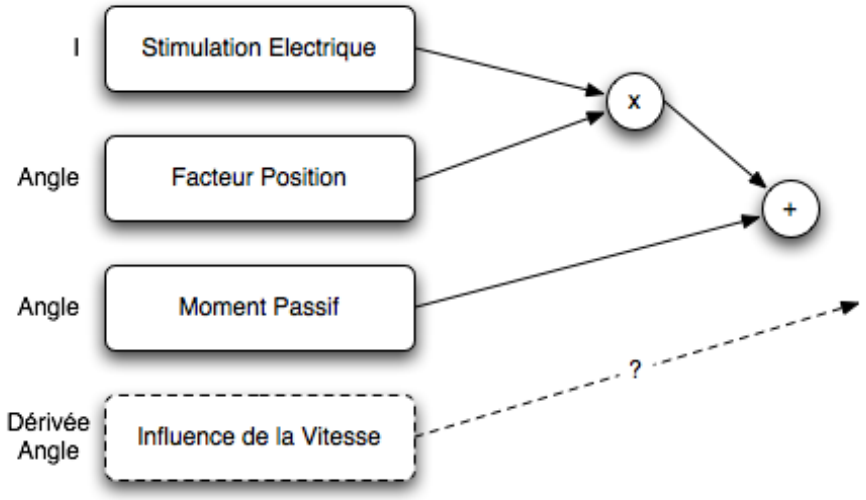


FIG. 5.2 – diagramme du modèle de muscle de P. Métrailler avec l'ajout possible de l'influence de la vitesse

$$
\begin{aligned}
\bar{M} &= a\Theta^3 + b\Theta^2 + c\Theta + d \\
M &= \texttt{interpolation du moment créé} \\
MomentMusculaire &= \bar{M} \times M
\end{aligned}
$$

$Avec$:
$\bar{M}$: `facteur d'atténuation du à la contraction/élongation du muscle`
M : `couple généré par la réaction du muscle à l'électro-stimulation`

FIG. 5.3 – Equation du moment musculaire

$$M(I) = M^- + (I - I^-)\frac{M^+ - M^-}{I^+ - I^-}$$

$Avec$:
$M(I)$: `couple généré par la stimualtion I`
I^+ : `Intensité supérieur connue la plus proche de I`
I^- : `Intensité inférieur connue la plus proche de I`
M^+ : `couple généré par la simulation du muscle avec` I^+
M^- : `couple généré par la simulation du muscle avec` I^-

FIG. 5.4 – Interpolation linéaire

Chapitre 6

Simulation avec l'orthèse du genou

6.1 Création de l'orthèse

L'orthèse du genou à été simulée comme un deuxième robot dans Webots et à été ajoutée au "monde". C'est donc un autre objet à part entière qui peut intéragir librement avec l'humanoïde. L'orthèse comporte un seul servomoteur qui est celui de son bras articulé. Le reste du modèle est composé de groupes de forme simples assemblées de sorte à représenter l'orthèse du genou.

Contrairement au modèle de l'humanoïde auquel un enrobage visuel avait été ajouté, la forme du robot reste celle des formes qui la composent. Des textures ont été ajoutées dans un souci esthétique. La figure 6.1 présente le résultat graphique du modèle.

Une vidéo fournie en annexe à ce rapport montre le robot humanoïde exercent. Dont le muscle du élecro-stimulé avec 35mA exerce un couple, dépendant de la position, sur le bras de l'orthèse. Le moteur de l'orthèse va se déplacer jusqu'à atteindre une position de 45 degrés puis se stabiliser.

6.2 Mesure du couple produit par le genou sur l'orthèse

Après avoir résolus les problèmes rencontrés qui sont résumés dans la section 6.3. La solution finalement adoptée est de mesurer le couple produit au niveau du joint entre le bras de l'orthèse et la fixation. La valeur dont on parle est donc la composante en Z du couple généré par le joint sur le bras de l'orthèse.

6.3 Problèmes rencontrés

Voici une liste des plus gros problèmes rencontrés dans cette partie du travail.

- Feedback : comme déjà mentionné dans la section la valeur contenue dans la structure Feedback, qui est utilisé pour lire les couples, ne tient pas compte des couples ajouté par l'utilisateur. Il a donc fallu tenir compte de ce paramètre pour choisir l'emplacement auquel la mesure est effectué (autrement dit sur quel corps).
- Oscillations : lors des premières mesures, le valeurs lues oscillaient fortement. Après avoir passé un moment à cherché dans les paramètre d'élasticité des corps (du à l'idée que la jambe rebondissait probablement sur la fixation), J'ai constaté que la fixation oscillait. La figure 6.2(a) présente le problem et la figure 6.2(b) montre la solution trouvée. En effet, même avec un pas de temps très petit la fixation plate oscille[1]. Ce problem est donc résolu en changeant simplement la forme de la fixation.
- Le probleme de masse cachée : pour simplifier les calculs de lecture de couple sur l'orthèse, il est agréable de modéliser la fixation comme le bras de l'orthèse avec une masse nulle. Webots permet de

[1] Ce problème semble dû à la façon dont est modélisé un contacte plat dans la librairie physique. Le nombre de points de contacte et limité, et le calcul de la réaction paraît inapproprié. Ce phénomène peut se produire pour tous les contactes plats : par exemple avec un cube posé le sol simulé avec un timestep un peut trop grand se met à osciller de la même manière

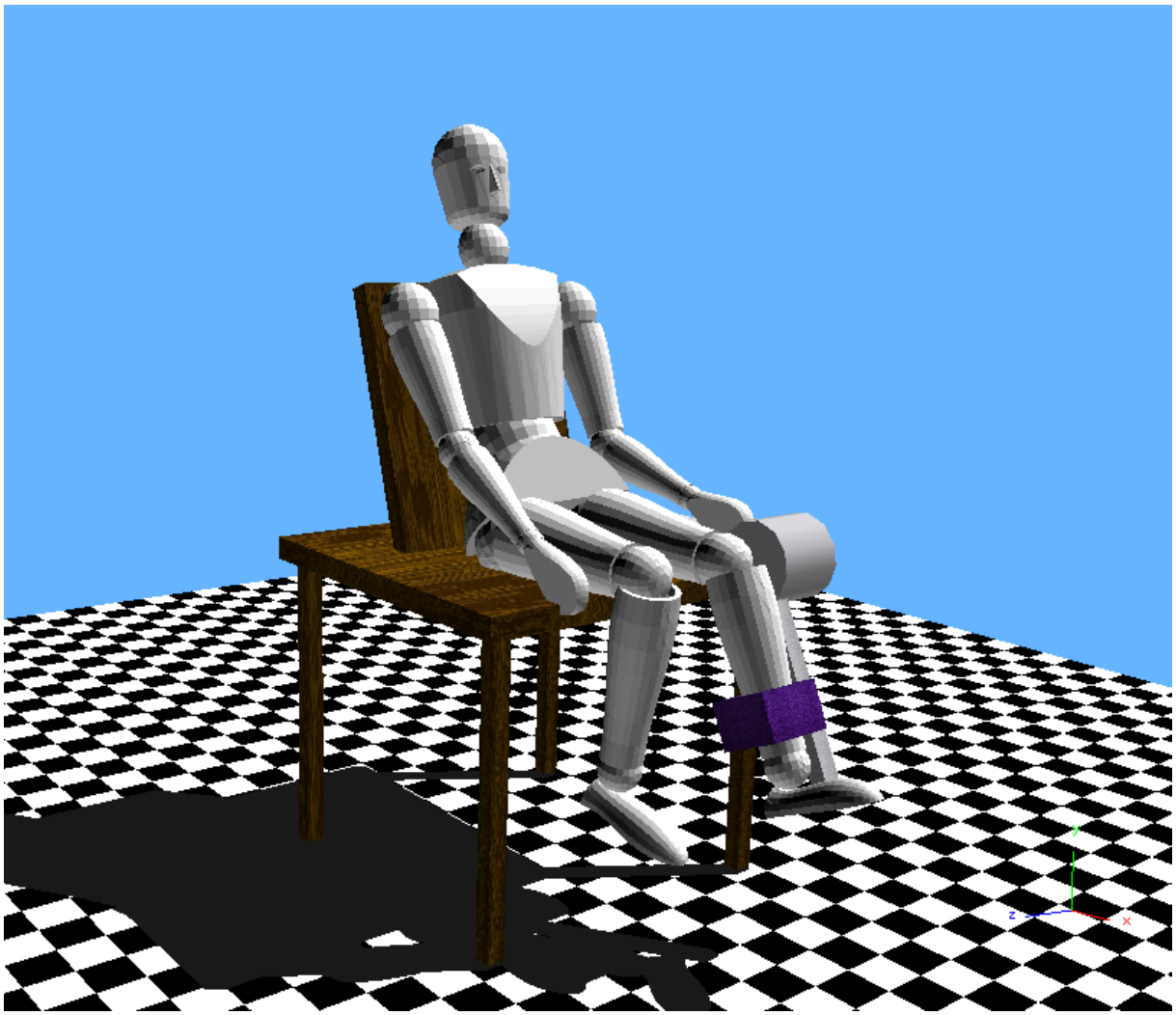
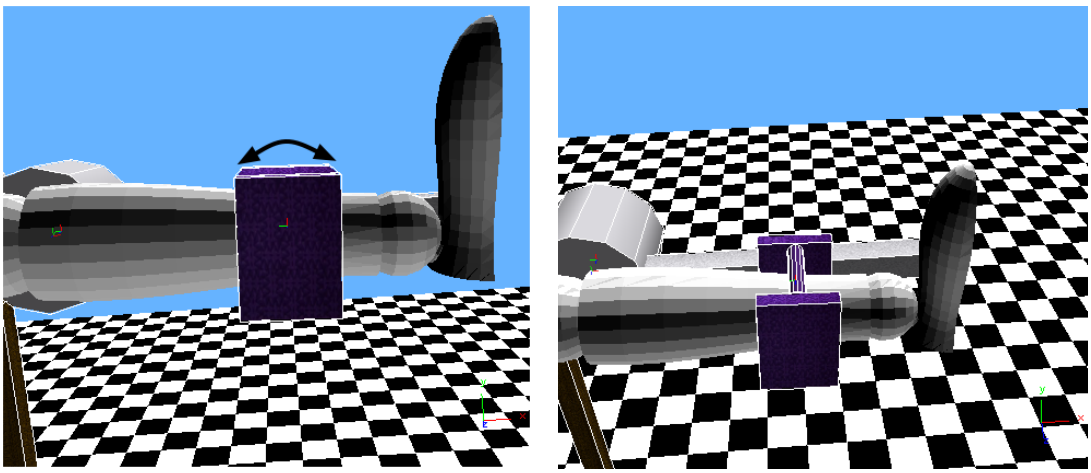


FIG. 6.1 – Modèles d'orthèse du genou avec humanoïde

spécifier la masse de deux façons : par l'intermédaire d'une densité ou en spécifiant la masse. Pour spécifier la masse on met le champs densité à -1 et le champs masse à la valeur désirée. Cependant, un objet avec une masse nulle n'est pas acceptée et, sans en avertir l'utilisateur, Webots[2] défini un objet de densité 1000. Si l'on veut une masse négligeable, il faut donc la spécifier comme très petite mais pas zéro.



(a) problème (b) solution

FIG. 6.2 – Illustration du problème d'oscillaitons du à la fixation et sa solution

6.4 Résultats

Cette section présente les résultats des expériences qui tentent mesure le couples que la jambe exerce sur le bras de l'orthèse.

6.4.1 Cas simple statique

Ces premiers tests visent à vérifier que la valeur de couple lue dans la simulation est correcte. Il se déroule de la manière suivante :

- On applique un couple constant tout au long de la simulation sur le genou (0 pour le test à vide).
- La simulation démarre a 90°(jambe fléchie), et va atteindre la position 0°(jambe tendue).
- Une fois stabilisié à la position finale (0°), on lit la valeur du couple exercée.

Sans couple (masse de la jambe)

La figure 6.3 schématise le calcul qui suit : On cherche à calculer le couple exercé sur l'articulation par la masse de la jambe quand celle-ci est tendue (à 0°).

définition : $\overrightarrow{\tau} = \overrightarrow{r} \times \overrightarrow{F}$

Avec :

- $\overrightarrow{r}$ le vecteur position du point par rapport auquel on calcul le couple jusqu'au point d'application de la force.
- $\overrightarrow{F}$ la force exercée au point d'application.

[2]ou la librairie physique, ceci n'est pas documenté

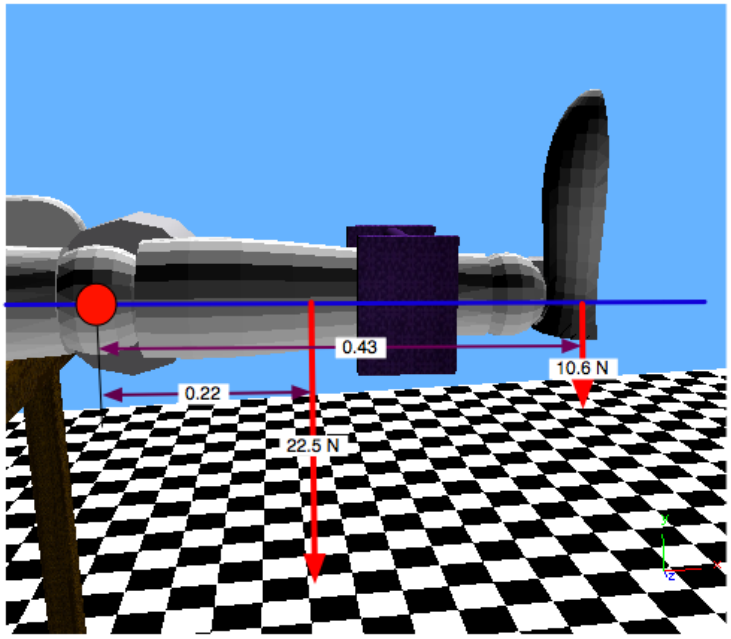


FIG. 6.3 – représentation des forces

dans notre cas

$$\begin{aligned} \overrightarrow{\tau}_{genouTotal} &= \overrightarrow{\tau}_{m_{tibia}\ sur\ genou} + \overrightarrow{\tau}_{m_{pied}\ sur\ genou} \\ &= \overrightarrow{r_1} \times \overrightarrow{F_1} + \overrightarrow{r_2} \times \overrightarrow{F_2} \\ &= \overrightarrow{r_1} \times (m_{tibia} * \overrightarrow{g}) + \overrightarrow{r_2} \times (m_{pied} * \overrightarrow{g}) \\ => \overrightarrow{\tau}_{genouTotal} &= 0.186 * (3.488 * 9.82) + (0.431 + 0.034) * (1.088 * 9.82) \\ &= 6.384 + 4.962 \\ a &= 11.346 Nm \end{aligned} \tag{6.1}$$

résultat de la simulation : 11.78 Nm

erreur : $\frac{(11.78-11.36)}{11.36} = 3.7\%$

Avec un couple

La table 6.1 donne les résultats des différentes simulations avec

Couple généré [Nm]	Couple mesuré [Nm]	Couple généré mesuré* [Nm]	erreur [%]
0	11.78	0	0
10	-1.4	10.38	3.8
100	92.02	103.8	3.8
200	195.79	207.57	3.785
500	507.76	519.54	3.908

* : Couple mesuré - Couple généré par la masse(obtenu précèdement)

TAB. 6.1 – Resultats de simulation de l'orthèse

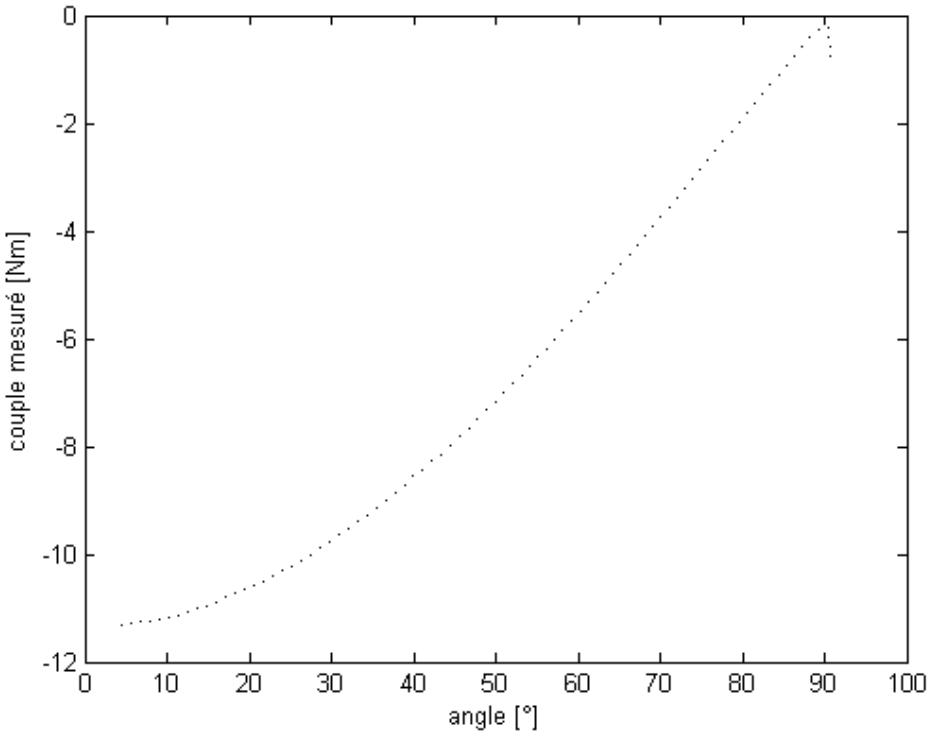


FIG. 6.4 – couple généré par la masse en fonction de la position

Erreur et sa source

Une erreur d'environ 3.8% est observée entre les valeurs lues et valeurs attendues. La valeur lue étant supérieur. L'erreur ne provient pas de Webots mais du fait qu'on lis le couple généré sur la bras de l'orthèse par rapport à son axe de rotation et non sur la jambe par rapport à l'axe du genou.

Un décalage entre les deux axes est donc vraisemblablement à l'origine de cette différence. Ce décalage pourrait être corrigé en ce point mais il faut garder à l'esprit que le robot est fixé à l'orthèse uniquement par son interaction avec la fixation (et la table sur laquelle il est assis). Le genou se déplace donc indépendamment de l'axe du moteur(qui lui devrait être fixe car rattaché au monde par un joint fixe). Ces deux axes sont donc alignés du mieux possible au départ de la simulation mais l'humanoïde pouvant bouger ses jambes il peuvent se trouver légèrement décalés après un moment de simulation.

De plus, cette erreur correspond plus ou moins à 1cm de décalage entre les deux axes. Ce type d'erreur est de plus un problème qui se présente aussi lors de mesures sur des patients réels.

6.4.2 Couple généré par la masse en fonction de la position

La figure 6.4 présente les couples mesurés sur l'axe z en fonction de la position de l'orthese alors qu'on ne génère aucun couple sur l'articulation. Le couple mesuré est donc uniquement celui provenant de la masse de la jambe.

Remarques : 90est l'angle pour lequel la jambe est vertical. La mesure pour la jambe tendue est donc celle tout à gauche sur le graphe (Ceci est aussi valable pour les autres mesures par rapport à l'angle du genou). De plus la valeur du couple mesurée est négative (car en négatif sur l'axe Z).

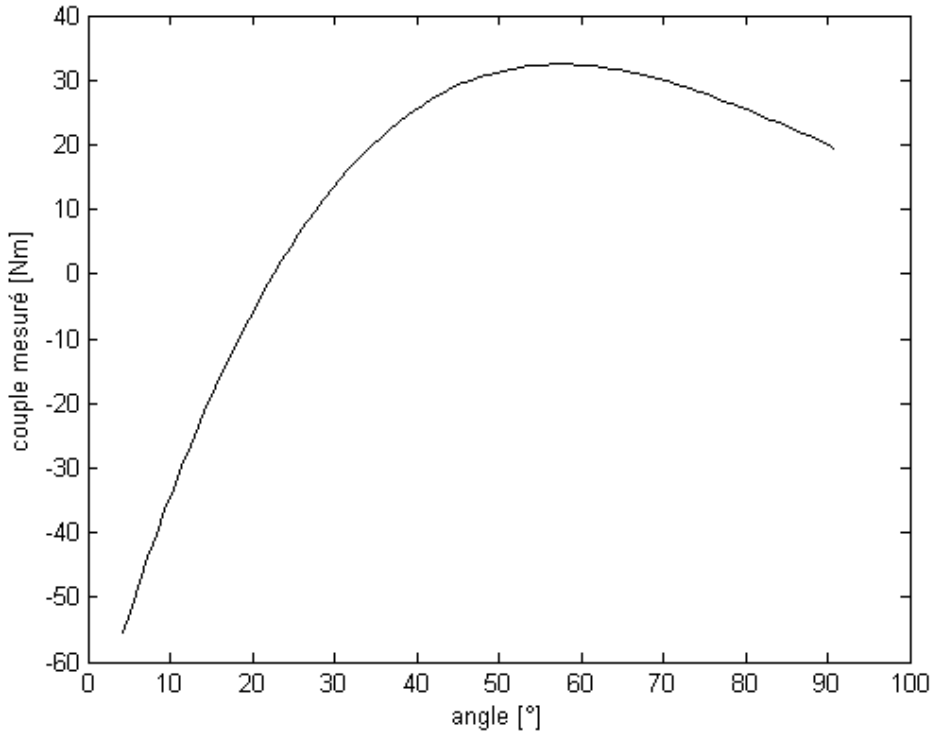


FIG. 6.5 – couple généré par le muscle, pour une stimulation électrique de 35mA, en fonction de la position

6.4.3 Couple généré par le muscle stimulé avec une intensité de 35 mA en fonction de la position

Les couples décris dans la figure 6.5 sont les couples résultants du couple généré sur le muscle pour une stimulation à 35mA. On voit sur la figure que l'intensité varie clairement en fonction de l'angle du muscle.

La figure 6.6 montre le couple mesuré avec stimulation et sans stimulation. De plus la courbe en traitillé est la superposition des deux autres. Cette dernière représente donc le couple généré par la muscle.

6.4.4 Données sur patient réels

Les données présentée par la figure 6.7 proviennent des résultat d'une expérience sur un patient. Ce sont ces résultats qui ont été utilisés pour modéliser le muscle. En effet les différents paramètres, comme par exemple la réaction à l'intensité de la stimulation électrique , diffèrent d'un patient à un autre.

La figure 6.8 montre l'approximation faite de l'influence de la position sur la réaction du muscle. On remarque sur 6.8(b) que en dehors des bornes de l'expérience réelle (20̊à 120̊) le polynôme peut prendre des valeurs absurde d'un point de vue réel. Ce d'ailleur ce qui fait que l'on observe une génération de couples négatif dans le cas où la jambe se trouve à moins de 20̊.

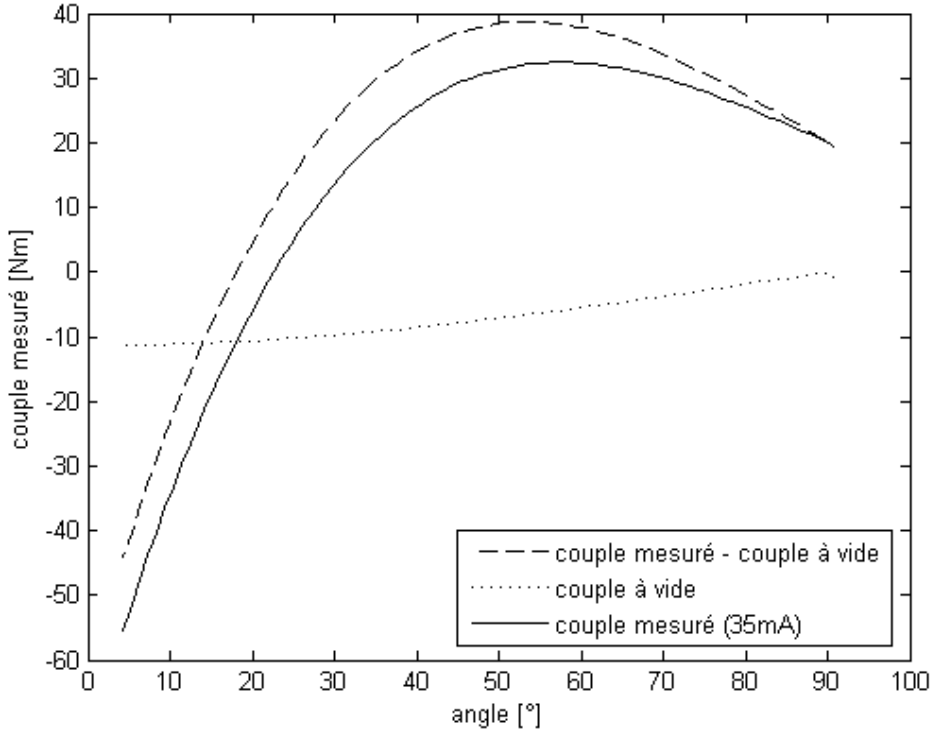


FIG. 6.6 – couples superposés

6.4.5 Comparaison entre couple mesuré et couple théorique généré

La figure 6.9 montre que l'on mesure la valeur prévue par le modèle. Un petit décalage de 3.8% semble être présent ici aussi du probablement à la légère différence entre les deux axes de rotation présentée dans la section 6.4.1.

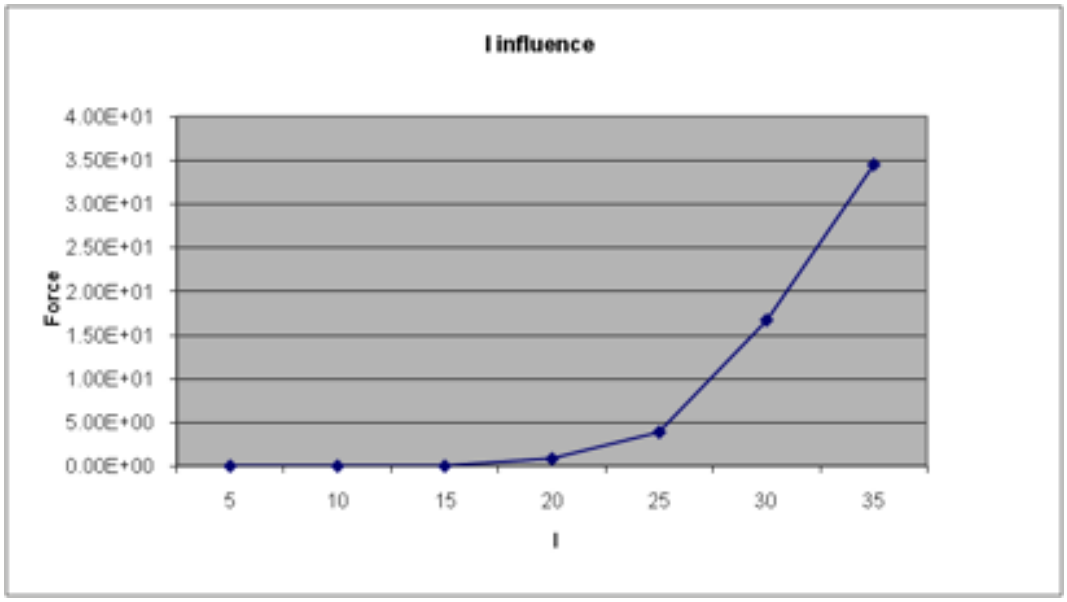
6.4.6 Analyse

Le couples part fortement dans le négatif quand l'angle passe en dessous de 20 °ceci est du au fait que le polynôme qui approche le facteur d'atténuation du couple passe en négatif. Ceci devrait être corrigé.

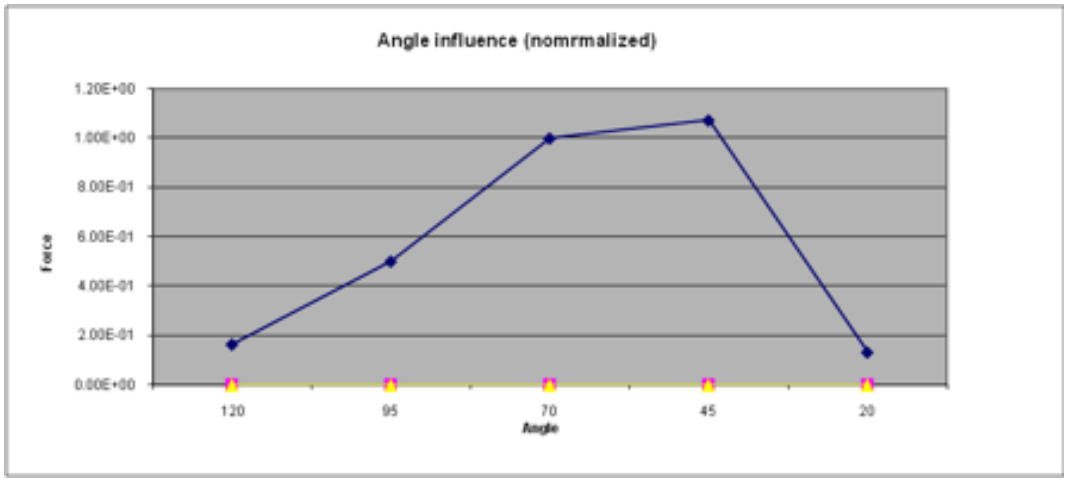
Mis à part le petit problème mentionné ci dessus, on remarque que les mesures sont en parfait accord le modèle implémenté. On peut donc être satisfait du comportement de Webots.

6.4.7 Remarque

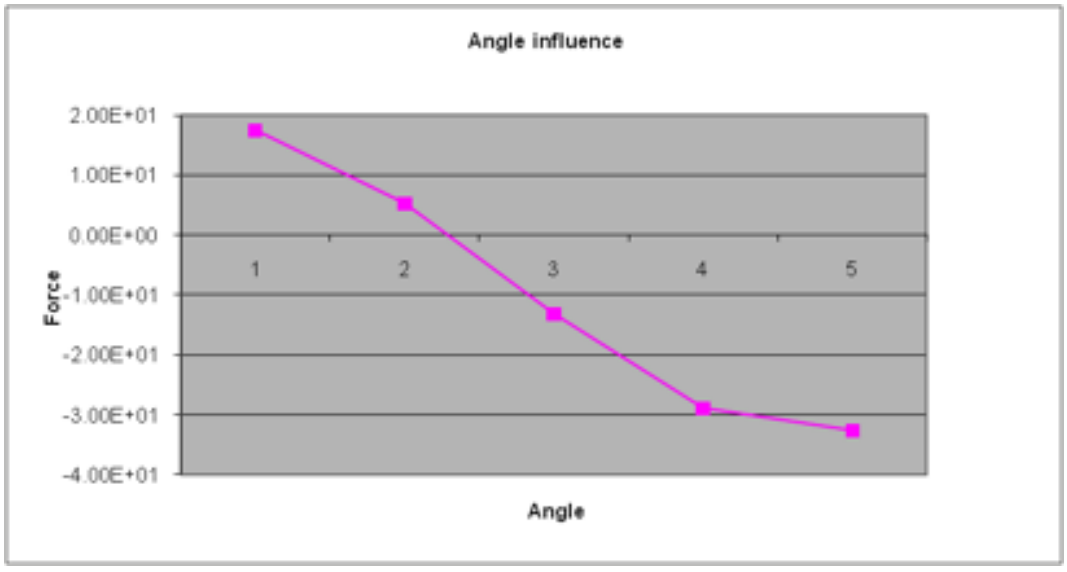
Le temps n'a malheureusement pas permit d'aller plus loin dans les simulations comme dans le rapport. Ce qui est regrettable car les résultats sont prometteurs mais manquent définitivement de comparaisons et d'analyse.



(a) Réaction du muscle par rapport à l'intensité électrique

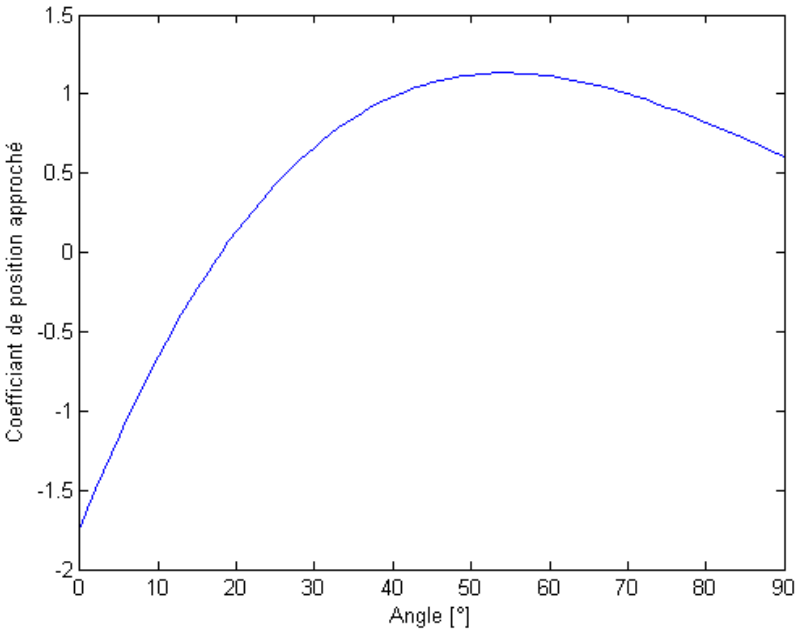


(b) Influence de la position (angle) normalisée

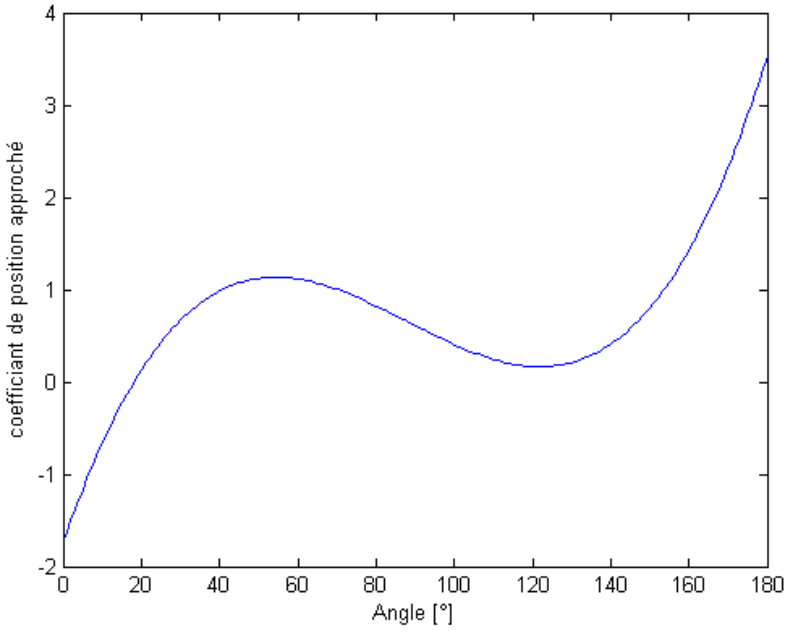


(c) Moment passif généré par le muscle et autres forces bio-méchaniques

FIG. 6.7 – Résultats d'expérience pour patient réel



(a) de 0° à 90°



(b) de 0° à 180°

FIG. 6.8 – Polynôme approchant l'influence de la position (figure 6.7(b))

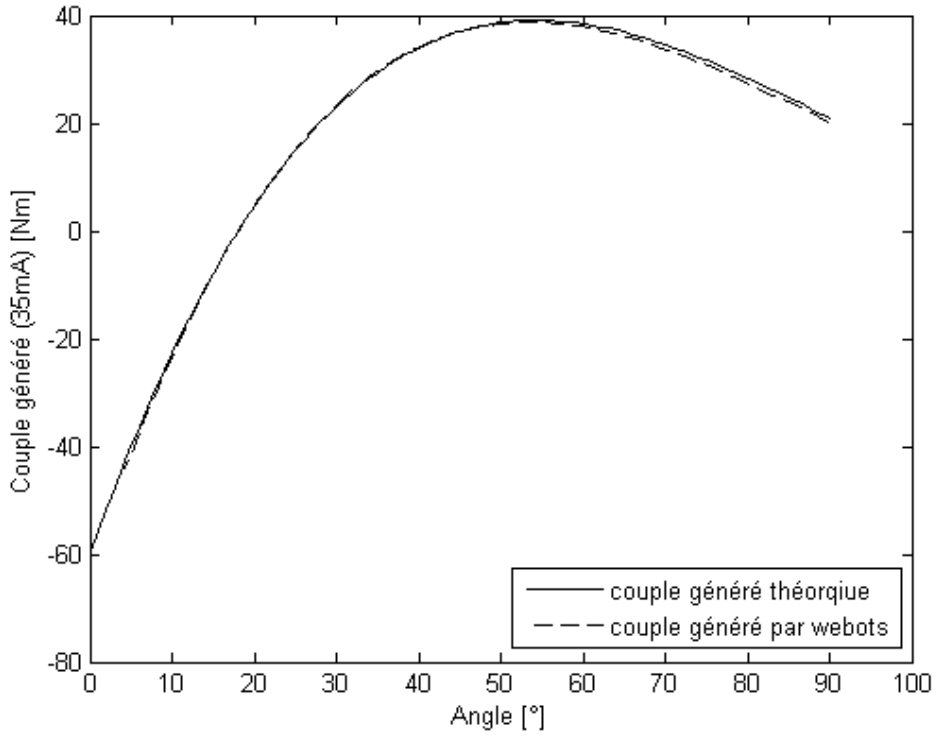


FIG. 6.9 – Supperposition du couple théorique généré et du couple mesuré sur l'orthèse

Chapitre 7

Améliorations et Travail futur

Ce projet d'investigation aux capacités de Webots à simuler un humanoïde et une orthèse de genou ouvre la porte à plusieurs travaux futurs ou simple à améliorations dont voici une liste non exhaustive :

- Si les mesures obtenues ne font que valider les valeurs obtenues expérimentalement ou par l'intermédiaires de modèles existant, Webots pourrait simuler des mouvements bien plus complexes et fournir des informations jusque là inconnues.
  Il serait donc interressant d'aller plus loin dans la simulation et de et de l'étendre au modèle du MotionMaker
- Le modèle de muscle actuel, peut déjà être étendu simplement en lui ajoutant la contribution de la vitesse, ce paramètre($\dot{\theta}$) est déjà calculé et ne demande qu'a être utilisé. De plus le moment passif pourrait être mieux modélisé car en ce moment le seul moment passif généré est celui de la gravité.
- Comme décris dans la section 3.4, le programme java de génération de modèles pourrait être amélioré pour pouvoir générer facilement des modèles différents (taille, masse, forces etc)
- Les textures, spécialement celles de l'orthèse du genou sont de taille assez grande et il est possible qu'elle ralentissent la simulation. Des textures plus optimisées pourraient augmenter la vitesse de simulation. Si on veut privilégiée la vitesse de simulation plutôt que l'aspect esthétique on pourra même supprimer les textures de l'orthèse ainsi que l'enrobage du robot.

Chapitre 8

Conclusion

Le but de ce projet était de vérifier que Webots est capable de simuler fidèlement un système aussi complexe que les membres inférieurs d'un humain en mouvement et de créer un tel modèle.

Le modèle à été crée et les résultats obtenus sont encouragent :

- Le chapitre 3 présente un humanoïde à l'aspect visuel réussi qui pourra comme espéré être un outil permettant de mettre en évidence certains mouvements dans le but de les présenter aux patients.
- Au chapitre 4, on voit que les couples mesurés lors de la marche sont de la même forme que ceux calculés par le modèle Matlab. Cependant, il faut nuancer cet enthousiasme, car l'intensité des deux valeurs diffère de manière inquiétante. Il y a plusieurs explications à cela, comme la différence de modèle utilisé par les deux représentations, mais cela ne permet honnêtement pas d'en conclure que les intensités mesurées sont celles qu'elles devraient être et il serait probablement nécessaire de passer plus de temps à comprendre ce problème avant de se lancer dans des expériences dynamiques.
- Le modèle de muscle, présenté au chapitre 5 et testé avec l'aide d'une orthèse de genou dans le chapitre 6, est un succès. Le temps n'a malheureusement pas permit d'aller très loin dans le rapport sur cette partie du projet. On notera cependant que les résultats obtenus sont ceux que le modèle prévoit.

Webots semble donc tout à fait capable de simuler correctement les interactions entre les différents membres et pourra probablement faire de même avec des problèmes plus complexes comme le modèle du MotionMaker.

Même si Webots fournit de bons résultats, il est incontestablement plus complexe, moins flexible et facile à modifier qu'un modèle Matlab.

Pour compliquer les choses on utilise Webots d'une manière différente de celle pour laquelle il est conçu. En effet alors que Webots est principalement conçu et utilisé pour modéliser des prototypes et tester des contrôleurs, l'utilisation faite de Webots dans le cadre de la modélisation de l'humanoïde est beaucoup plus orientée vers des mesures physiques sur le modèle. L'accès et l'utilisation de la librairie physique s'avèrent être moins documentés et plus complexes que l'utilisation "classique" de Webots.

Ceci pose la question de l'utilité d'un tel modèle : Quelles sont les personnes susceptibles de l'utiliser ?

D'un point de vue personnel, j'ai beaucoup apprécié ce travail. Il est multidisciplinaire dans le sens qu'il touche à la biologie,à la mécanique,à la physique et bien entendu à l'informatique. De plus il porte sur un problème concret et il était toujours très agréable de pouvoir se repérer à la réalité qui est derrière la simulation.

Ce projet a aussi présenté son lot de problèmes et a été accompagné de moments frustrants. La progression a été plus fastidieuse qu'imaginé, interrompue par les divers problèmes mentionnés dans ce rapport qui se sont avérés plutôt difficiles à résoudre de par le manque de documentation. Je tiens donc à remercier les assistants et professeurs qui m'ont beaucoup aidé à comprendre et à résoudre ces problèmes.

Bibliographie

[CLV92] Jermy O'Connor Christopher L Vaughan, Brian L Davis. Dynamic of Human Gait. Number 0-620-23558-6. Kiboho Publishers, 2nd edition, 1992.

[con97] Web3D consortium. Vrml97 functional specification. International Standard ISO/IEC 14772-1, 1997.

[LSR] LSRO. Cyberthèse. http ://lsro.epfl.ch/page62460.html.

[Ltda] Cyberbotics Ltd. Webots reference manual.

[Ltdb] Cyberbotics Ltd. Webots user guide.

[Mét05] Patrick Métrailler. Système robotique pour la moilisation des membres inférieurs d'une personne paraplégique. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2005.

[Mic04] O. Michel. Webots : Professional mobile robot simulation. Journal of Advanced Robotics Systems, 1(1) :39–42, 2004.

[Smi06] Russel Smith. Open dynamic engine, user guide. 2006.

[SotISoBI02] The Standardization and Terminology Committee (STC) of the International Society of Biomechanics (ISB)". Isb recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part i : ankle, hip, and spine. Journal of Biomechanics, 2002.

[Web] Webots. http ://www.cyberbotics.com. Commercial Mobile Robot Simulation Software.

[Win90] David A. Winter. Biomechanics and motor control of human movment. Number 0-471-509008. Wiley-Interscience, 1990.

Chapitre 9

Annexes

Ce rapport est rendu avec deux compléments :
- 2 exemplaires du résumé.
- un CD-ROM dont le contenu est détaillé dans la section suivante.

9.1 Contenu du CD-ROM

Le CD-ROM rendu contient quatre dossiers principaux :

1. report : partie rapport qui contient le rapport en pdf, son fichier source en format Lastex et les illustrations du rapport.
2. doc : une partie de la documentation qui a été utile durant le travail et qui pourrait avoir un intérêt pour le lecteur ou la personne qui réutilise ce travail.
3. model : La partie logiciel du travail qui contient les différents modèles, contrôleurs, plugins physiques etc ... Le détail de la partie logiciel est donnée dans le fichier README.txt contenu dans ce dossier.

Chapitre 10

Outils de travail

Webots 5.10
Matlab 7.1
Visual Studio Express 2006
Eclipse 3.2 Java 1.5
Latex / Mac OSX

Chapitre 11

Appendice

	a_0	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4
xbassin	-0.006201	3.055	0.01583	-1.183	-0.02031	3.469	0.01803	0.164	0.005529
ybassin	0.3178	4.649	-0.1403	0.2473	-0.02055	1.139	-0.2308	-0.9058	0.1758
zbassin	10.07	-0.06378	0.09167	-0.003972	-0.002268	-0.4942	0.07205	0.0292	0.0195
xhanche	0.03532	1.577	-0.5737	-1.756	-0.08828	5.499	-0.6867	0.3326	0.1191
yhanche	-1.712	-4.553	-1.918	-0.6158	-1.046	-0.1401	-0.1449	0.4175	0.6809
zhanche	6.46+14.88	22	-1.168	-0.9951	-0.002272	3.181	-1.981	0.5826	0.235
genou	25.37	-1.873	-12.11	-0.3987	0.0211	-17.22	10.7	3.995	0.1302
cheville	3.614	0.4274	0.3072	-3.106	1.63	6.633	-6.114	1.088	-0.3764

TAB. 11.1 – paramètres de Fourrier

total	tronc	piedY	piedX	piedZ	tibia	cuisse	hanchesZ	colonne	épaulesZ	épaules	bras	avant bras	main	tête
1	0.733	0.039	0.152	0.055	0.246	0.245	0.191	0.406	0.258	0.085	0.186	0.146	0.108	0.13
2	1.466	0.078	0.304	0.11	0.492	0.49	0.382	0.812	0.516	0.17	0.372	0.292	0.216	0.26
1.75	1.28275	0.06825	0.266	0.09625	0.4305	0.42875	0.33425	0.7105	0.4515	0.14875	0.3255	0.2555	0.189	0.2275

TAB. 11.2 – tailles segments [m]

masse totale	pied	tibia	cuisse	hanches	tronc	bras	avant-bras	main	tête
1	0.0145	0.0465	0.1	0.142	0.355	0.028	0.016	0.006	0.081
75	1.0875	3.4875	7.5	10.65	26.625	2.1	1.2	0.45	6.075

TAB. 11.3 – masses segments [Kg]